This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04L 12/16 — A1

(11) International Publication Number: WO 00/60809

(43) International Publication Date: 12 October 2000 (12.10.00)

(21) International Application Number: PCT/US00/08688

(22) International Filing Date: 30 March 2000 (30.03.00)

(30) Priority Data:

60/127,364	1 April 1999 (01.04.99)	US
60/137,888	7 June 1999 (07.06.99)	US
09/358,878	22 July 1999 (22.07.99)	US

(81) Designated States: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

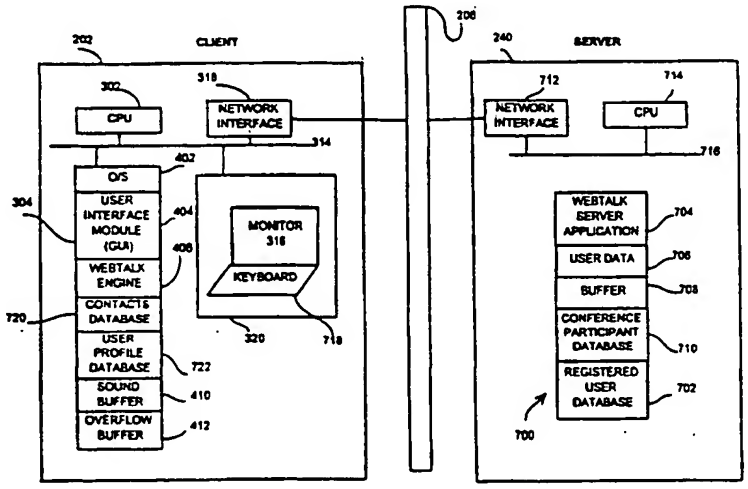
Published
With international search report.
With amended claims.

(71) Applicant: MULTITUDE, INC. [US/US]; Suite 1050, 601 Gateway Boulevard, South San Francisco, CA 94080 (US).

(72) Inventors: MORRIS, James, E., G.; 1136 Lincoln Court, San Jose, CA 95125 (US). LERNER, Edward, A.; 2833 Union Street, San Francisco, CA 94123 (US). POELKE, James; 826 E. Santa Inez, San Mateo, CA 94401 (US). MIN, Arthur; 308 Gateway Drive #236, Pacifica, CA 94404 (US). MONEY, William; 2301 Spring Creek Drive, Austin, TX 78704 (US).

(74) Agents: GALLIANI, William, S. et al., Pennie & Edmonds LLP, 1155 Avenue of the Americas, New York, NY 10036 (US).

(54) Title: APPARATUS AND METHOD FOR ESTABLISHING AN AUDIO CONFERENCE IN A NETWORKED ENVIRONMENT



(57) Abstract

The present invention provides a graphical user interface (GUI) (404) and network based telephony system (206) and method for establishing an audio conference. In a network environment where a plurality of computers connected to allow audio/text conferences and peer-to-peer text messaging. Once a conference is created, each participant may add a new participant to the conference at any time during the conference call by sending a call request to the computer of the new participant. All participants in an audio conference may freely speak at any time they want without losing information or audio data from other participants. Each of the participants in an audio conference is in audio communication with other participants. This functionality is achieved by combining a buffer control mechanism and an audio conference facility (710). The invention also allows peer-to-peer text/audio messaging between the computers in the network and overcomes firewall incompatibility problems.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

BNSDOCID: <WO___0060809A1_I_>

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 October 2000 (12.10.2000)

PCT

(10) International Publication Number
WO 00/60809 A1

(51) International Patent Classification[7]: H04L 12/16

(21) International Application Number: PCT/US00/08688

(22) International Filing Date: 30 March 2000 (30.03.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/127,364	1 April 1999 (01.04.1999)	US
60/137,888	7 June 1999 (07.06.1999)	US
09/358,878	22 July 1999 (22.07.1999)	US

(71) Applicant: MULTITUDE, INC. [US/US]; Suite 1050, 601 Gateway Boulevard, South San Francisco, CA 94080 (US).

(72) Inventors: MORRIS, James, E., G.; 1136 Lincoln Court, San Jose, CA 95125 (US). LERNER, Edward, A.; 2833 Union Street, San Francisco, CA 94123 (US). POELKE, James; 826 E. Santa Inez, San Mateo, CA 94401 (US). MIN, Arthur; 308 Gateway Drive #236, Pacifica, CA 94404 (US). MONEY, William; 2301 Spring Creek Drive, Austin, TX 78704 (US).

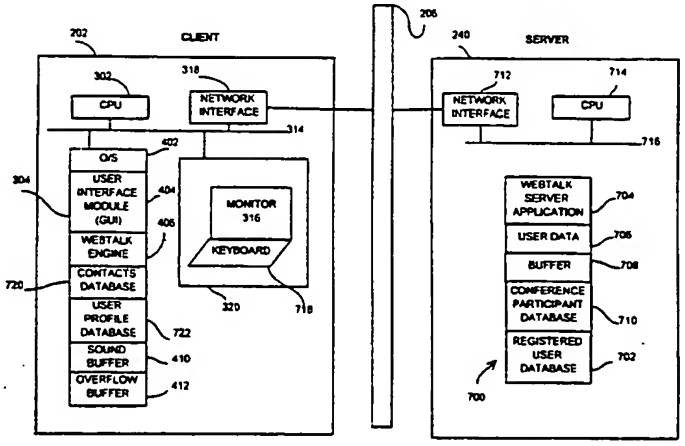
(74) Agents: GALLIANI, William, S. et al.; Pennie & Edmonds LLP, 1155 Avenue of the Americas, New York, NY 10036 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU,

[Continued on next page]

(54) Title: APPARATUS AND METHOD FOR ESTABLISHING AN AUDIO CONFERENCE IN A NETWORKED ENVIRONMENT



(57) Abstract: The present invention provides a graphical user interface (GUI) 404 and network based telephony system 206 and method for establishing an audio conference. In a network environment where a plurality of computers connected, the invention establishes a connection between plurality of computers, to allow audio/text conferences and peer-to-peer text messaging. Once a conference is created, each participant may add a new participant to the conference at any time during the conference call by sending a call request to the computer associated with the new participant. All participants in an audio conference may freely speak at any time they want without losing information or audio data from other participants. Each of the participants in an audio conference is in audio communication with other participants. This functionality is achieved by combining a buffer control mechanism and an audio conference facility 710. The invention also allows peer-to-peer text/audio messaging between the computers in the network and overcomes firewall incompatibility problems.

WO 00/60809 A1

MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*
— *With amended claims.*

(48) Date of publication of this corrected version:
15 March 2001

(15) Information about Correction:
see PCT Gazette No. 11/2001 of 15 March 2001, Section II

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/08688

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :H04L 12/16
US CL :370/259, 260, 261, 262

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 370/259, 260, 261, 262

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

none

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EAST, WEST
search terms: conference, GUI, peer-to-peer, list

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,363,507 A (NAKAYAMA et al) 08 November 1994, col 10, lines 62 +.	1-17
X	US 5,491,743 A (SHIIO et al) 13 February 1996 see fig 4.	1-17
A,P	US 6,011,782 A (DESIMONE et al) 04 January 2000, see fig 3.	1-17
A,P	US 5,930,238 A (NGUYEN) 27 July 1999, see fig 6b.	1-17

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents.	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search	Date of mailing of the international search report
02 JULY 2000	25 JUL 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer ANDREW LEE Telephone No. (703) 305-1500

Form PCT/ISA/210 (second sheet) (July 1998) *

BNSDOCID: <WO___0060809A1_I_>

APPARATUS AND METHOD FOR ESTABLISHING AN AUDIO CONFERENCE IN A NETWORKED ENVIRONMENT

This application claims priority to United States Provisional Patent Application Serial No. 60/127,364, filed April 1, 1999, entitled "Internet Telephony System and Method," by James E. G. Morris and Edward A. Lerner, and to United States Provisional Patent Application Serial No. 60/137,888, filed June 7, 1999, entitled "Interface for a Voice-Over-Network Application," by Edward A. Lerner and Arthur W. Min.

BRIEF DESCRIPTION OF THE INVENTION

The present invention relates generally to audio communication. More specifically, the present invention relates to a network application for placing and receiving audio calls and conducting multi-party audio conferences in a networked environment.

CROSS-REFERENCE TO RELATED DOCUMENTS

The present invention is related to the subject matter disclosed in U.S. Patent No. 5,764,900 ("System and Method for Communicating Digitally-Encoded Acoustic Information Across a Network between Computers"), U.S. Provisional Patent Application Serial No. 60/127,364 ("Internet Telephony System and Method"), U.S. Provisional Patent Application Serial No. 60/137,888 ("Interface for a Voice-Over-Network Application"), and U.S. Patent Application Serial No. _____ ("Apparatus and Method for Creating Audio Forums") filed July 22, 1999. Each of these documents is assigned to the assignee of the present application and incorporated herein by reference.

BNSDOCID: <WO___0060809A1_I_>

BACKGROUND OF THE INVENTION

Conventional telephone and conference calls are made using Public Switched Telephone Networks (PSTNs) and/or commercial wireless networks. More recently, telecommunications services have begun to view the Internet as an alternative, less costly method for transmitting audio calls. The art of developing systems that enable two or more persons to connect through the Internet and conduct real-time conversations is often referred to as "Internet telephony."

Prior art Internet telephony systems suffer from a number of disadvantages when compared to PSTN-based telephony systems. First, many prior art systems are difficult to configure and use, particularly for users with limited technical experience. Second, prior art telephony systems are frequently incompatible with network firewalls that are used to restrict access to and from network resources. Third, many Internet telephony systems do not offer multi-party conferencing capability.

Further, prior art Internet telephony technology makes it difficult to implement a computer graphical user interface ("GUI") for multi-party conferencing. For example, FIG. 1 illustrates a prior art web browser-based voice conferencing environment. The user ace885 is connected to a specific chat room (site) in order to conduct a voice conference with users ned11111 and tinitina99. The current "chatter" is ace885, as indicated by the "ON-AIR" sign next to ace 885. When ace885 is "ON-AIR," other conference participants can not speak. When ace885 releases the communication channel, either ned11111 or tinitina99 may speak, depending upon which conference participant secures the channel first by pressing the "READY" button.

In this connection, conference participants may secure the communication channel and speak by toggling their "ON-AIR" button to "READY." The conference participant who presses the "READY" button first is granted access to the channel and is allowed to speak. Thus, a prior art computer GUI for multi-party conferencing represents a somewhat limited and inconvenient option to its user, which creates undesirable race conditions.

Moreover, it is possible to lose information in prior art Internet telephony systems. As an illustration, more than two people may race to press the "READY" button in an attempt to secure the communication channel at the same time.

EFSDOCID: <WO__0060809A1_I_>

Unfortunately, anything which might be spoken by the person who lost the race to secure the communication channel will not be heard by other conference participants and is consequently lost.

Also, in prior art Internet telephony systems, the conversation in a conference call is broadcast to all participants unless blocking is done using option menus. Unfortunately, prior art blocking requires a user to execute blocking every time the user joins a conference.

In view of the foregoing, it would be highly desirable to provide an improved Internet telephony system. In particular, it would be highly desirable to provide an improved GUI for conducting multi-party audio conferences.

SUMMARY OF THE INVENTION

The present invention provides a GUI and network-based telephony system and method for establishing an audio and/or text conference. In a network environment where a plurality of computers are connected, the invention generates a call request in response to the call request of a first user who is connected to one of the networked computers. The computer to which the first user is connected transmits the call request to a second computer on the network, notifying the second computer of the call request and the identity of the first user. When a second user connected to the second computer accepts the call request from the first user, the present invention establishes a connection between the first and second computers. Once a connection is established between the first and second users, the users may conduct an audio and/or text conference and/or engage in peer-to-peer messaging.

The GUI of the invention displays the participants in a conference and provides a contacts list. On the GUI, each participant in the conference can see a contacts list and a user list, add/find a user, and initiate a text conference by engaging a button on the GUI. A participant in an audio or text conference may add a new participant to the audio or text conference at any time during the conference by sending a call request to the computer of the new participant. In an embodiment of this invention, an audio conference participant need only select a name on the contacts list with a single mouse click to initiate a call request to the computer of the new participant. If the new participant accepts the call request, all audio or text communication originating from

BNSDOCID. <WO__0060809A1_I_>

the conference participants is directed to the new participant depending on the type of conference being created. Thereafter, all audio or text communication originating from the new participant is broadcast to other conference participants.

The invention is highly advantageous because all participants in an audio conference may freely speak at any time they want without losing information or audio data from other participants. Each of the participants in an audio conference is in audio communication with other participants. This functionality is achieved by combining a buffer control mechanism and an audio conference facility. The computers in the network utilize a buffer and a buffer control mechanism in order to eliminate overflow conditions and reduce audio data loss.

In addition, the invention allows direct peer-to-peer messaging between the computers in the network. If a client computer with which a peer-to-peer messaging is desired is behind a firewall, the present invention enables that client computer to route messages through an audio server, thereby overcoming firewall incompatibility problems.

BRIEF DESCRIPTION OF THE DRAWINGS

For a better understanding of the invention, reference should be made to the following detailed description taken in conjunction with the accompanying drawings, in which:

Figure 1 illustrates a typical prior art audio conferencing system based on a web browser.

Figure 2 illustrates the general architecture of the multi-party conference system 200 according to one embodiment of the present invention.

Figure 3 illustrates components of an exemplary client computer 202 that may be used in accordance with one embodiment of the invention.

Figure 4 illustrates a memory 304 of a client computer utilized in accordance with one embodiment of the invention.

Figure 5(A) illustrates the structure of an audio data packet used in accordance with one embodiment of the invention.

BNSDOCID: <WO___0060809A1_I_>

Figure 5(B) illustrates an operation of the overflow buffer 412, the sound buffer 410, and the sound mixer 408 in greater detail in accordance with one embodiment of the invention.

Figure 6 illustrates functional components of a webtalk engine 406 utilized in accordance with one embodiment of the invention.

Figure 7 illustrates a client computer and an audio server connected through a network in accordance with one embodiment of the invention.

Figure 8 illustrates a GUI of an audio communication application provided in accordance with one embodiment of the present invention.

Figure 9 illustrates processing steps for an audio communication application that may be executed in accordance with one embodiment of the invention.

Figure 10 illustrates an example of a "Find User" window provided in accordance with one embodiment of the invention.

Figure 11 illustrates an example of a user option window 1101 provided in accordance with one embodiment of the invention.

Figure 12 illustrates processing steps for creating a block list for a user in accordance with one embodiment of the invention.

Figure 13 illustrates a login window 1301 provided in accordance with one embodiment of the invention.

Figure 14 illustrates processing steps for connecting to a server for a conference in accordance with one embodiment of the invention.

Figure 15 illustrates an example of a broadcast window 1501 opened by user CNF_Room for text conferencing.

Figure 16 illustrates a text messaging window 1601 activated in accordance with one embodiment of the invention.

Figure 17 illustrates processing steps performed by a server for transmission of audio data to client computers in accordance with one embodiment of the invention.

Figure 18 illustrates processing steps of a client computer for receiving audio data in accordance with one embodiment of the invention.

Figure 19 illustrates an alternate embodiment of processing steps for receiving audio data for a client computer.

BNSDOCID: <WO___0060809A1_I_>

Figure 20 illustrates an alternate embodiment for processing steps for controlling audio data transmission.

Figure 21 illustrates processing steps for conducting multiple calls and conferences for a client computer in accordance with one embodiment of the invention.

Figure 22 illustrates processing steps of a server for establishing a peer-to-peer messaging session in accordance with one embodiment of the invention.

Like reference numerals refer to corresponding parts throughout the drawings.

DETAILED DESCRIPTION OF THE INVENTION

FIG. 2. illustrates the general architecture of a multi-party conference system 200 according to one embodiment of the present invention. In the system 200. computer devices are coupled for intercommunication via a public or private network such as the Internet. The computer devices include a plurality of client computers that are connected to a network 206 through any one of a number of interfaces. For example. a first client computer 202 is connected to the network 206 via either a wireless or dial-up modem connection to a network service provider 203 such as a commercial Internet Service Provider (ISP). a second client computer 204 connects to the network 206 via a T-1 line interface. and other client computers 210-220 connect to the network 206 via first and second Local Area Networks (LANs) 230 and 232.

The LANs may be protected with conventional firewalls 234 and 236 to inspect incoming packets and reject those originating from unauthorized or suspicious addresses and/or to limit outgoing packets. Other client computers may be connected to the network 206 via alternative interfaces. It will be appreciated by one skilled in the art that while a total of eight client computers are shown in FIG. 2. the present invention may be practiced using any number of client computers. There is at least one user associated with each of the client computers. The network 206 represents any suitable electronic media through which a plurality of computers may be connected. such as the Internet.

The system 200 shown in FIG. 2 also comprises audio servers 240-1 to 240-N coupled to the network 206. preferably via a high-bandwidth interface. The exact number of servers used in FIG. 2 is not important and any number of servers may be

BNSDOCID: <WO__0060809A1_I_>

used in accordance with the technology of the present invention. Each of the audio servers 240 comprises a computer having at least one processor, an interface to the network 206, and a memory (primary and/or secondary) for storing program instructions and other data. The audio servers 240 are collectively operative to enable each one of the client computers to conduct audio communications with any other client computers. Each audio server is specifically operative to initiate calls/conference connections responsive to a connection request received from an associated client computer, to determine the appropriate call architecture and protocol based on predetermined parameters, and to route audio data to one or more selected recipient client computers.

A database system 250, typically comprising one or more server computers, may also be coupled to the network 206 and may be configured to perform certain operations such as authentication of users desiring to utilize the services of the telephony system. In alternate embodiments of the invention, it is also possible that client, server, and database functions may be combined in a single computer. Alternatively, any two of the client, server, and database functions may be combined and implemented on a single computer.

FIG. 3 illustrates components of an exemplary client computer 202. It is noted that FIG. 3 is intended to serve simply as a conceptual representation of the client computer, rather than illustrate an actual computer architecture. For example, those skilled in the art will recognize that multiple buses (rather than a single bus 314, as depicted) are typically employed to enable communication between the several components.

The client computer 202 includes a CPU (Central Processing Unit) 302 for executing instructions and controlling communication between and among the various components of the computer 202, and a memory 304 for storing data such as program instructions or text files. One or more non-volatile storage devices 306, such as floppy drives, hard disks and the like, may be operable to store and retrieve applications, data and text files, and other information. The client computer 202 is also provided with a loudspeaker 310 for converting to sound electrical signals representative of the audio transmissions of users of other client computers. The client computer 202 may additionally be provided with either a microphone 308 for generating electrical signals

BNSDOCID: <WO___0060809A1_I_>

representative of the speech of the associated user or some other audio signal 313. The microphone 308. loudspeaker 310. and audio signal 313 may be connected to the client computer through a sound card 312. which performs the requisite analog-to-digital and digital to analog signal conversion. Alternatively. the CPU 302 can perform these functions. Also coupled to the client computer 202 are a display device 316. a network interface 318 (such as an Ethernet card or modem) to enable connection of the computer 202 to the network. either directly or indirectly. and additional input/output devices 320. such as a keyboard. mouse. or printer.

FIG. 4 illustrates a typical structure of memory 304 of the client computer 202. The memory 304 comprises an operating system 402. a user interface module 404. a webtalk engine 406. a sound mixer 408. a sound buffer 410. and. optionally. an overflow buffer 412. The operating system 402 is operative to allocate memory and perform other low-level functions of the client computer 202. The user interface module 404 may comprise. for example. a conventional browser or audio communication application. and is preferably provided with a GUI that enables a user to interact with the client computer 202 by entering text. selecting icons. or performing similar operations. The browser in the user interface module 404 can be any of the commercially available browsers such as Netscape Communicator. Internet Explorer. or Opera. The audio communication application may run in a Microsoft Windows environment. although other platforms are within the scope of the invention. The operation of the audio communication application is described in U.S. Patent No. 5.764.900 and U.S. Provisional Patent Application No. 60/127.364.

The webtalk engine 406 enables a user to send and receive real-time audio communications to and from one or more remote users associated with other client computers. The webtalk engine 406 is preferably implemented as an ActiveX control or other form of linked library. The sound mixer 408 is used for mixing acoustic signals. In one embodiment of the invention. the sound mixer 408 is implemented by using a Microsoft operating system call. In an alternate embodiment of the invention. the sound mixer 408 is implemented directly on an audio card. It will be appreciated that numerous other implementations are possible for the sound mixer 408.

The sound buffer 410. which may comprise one 410-1 or more 410-N individual sound buffers. is used to store acoustic signals received from other client

BNSDOCID: <WO__0060809A1_I_>

computers. The overflow buffer 412, which may comprise one 412-1 or more 412-N individual overflow buffers, may be used to store overflow acoustic signals when the sound buffer 410 is unable to accept additional audio data.

In a preferred embodiment of the invention, the sound buffers of a client computer continue to convert their contents to sound and play it to the user regardless of whether the user is speaking. In the preferred embodiment, the audio data stored in four buffers are mixed together, converted to sound and presented to the user so that the user can hear other users speaking at the same time that the user is speaking. This embodiment eliminates the need to pause conversion of the buffer contents into sound when the user is speaking.

In an alternate embodiment of the invention, the conversion of the contents of the sound buffers to sound is paused by the client computer when the user of the client computer starts speaking. When the user stops talking, the sound buffers resume converting their content to sound. Optionally, pauses in human speech may be compressed. The effect of the buffer pause is that the user does not miss what other people are saying while the user was talking. The processing of current audio data is completed after the client computer finishes conversion of the buffer contents to sound.

FIG. 5(A) illustrates the structure of a typical audio data packet used in accordance with one embodiment of the invention. The packet 500 shown in FIG. 5(A) comprises a packet header 502 and payload 504. The packet header 502 has network protocol information such as the TCP (Transmission Control Protocol) or UDP (User Datagram Protocol), and IP (Internet Protocol) packet header information. In a preferred embodiment, the payload 504 is formatted to specify a type of packet, a start/middle/end-of-data indicator, the identity of the user who originated the packet, the packet data size, the packet sequence number, and one or more flag bytes. The information contained in the packet payload 504 is used by a recipient client computer to determine the size, origin, and sequence number of the incoming packet. In alternate embodiments of the invention, the identity of the user who originated the packet is not included in the payload 504, and the recipient computer relies on the information contained in the header 502 to determine the source of the packet.

BNSDOCID: <WO___0060809A1_I_>

FIG. 5(B) illustrates an operation of the overflow buffer 412, the sound buffer 410, and the sound mixer 408 in greater detail in accordance with one embodiment of the invention. A packet control module 508 of a client computer (for example 202) is coupled to the network interface 318 such as an Ethernet card or modem, to the sound buffer 410, and to the overflow buffer 412. In the embodiment shown in FIG. 5(B), the sound buffer 410 is implemented by sound buffers 410-1 through 410-N.

The packet control module 508 receives incoming audio packets and determines where they should go. For example, the packet control module 508 routes an incoming packet from other client computer to the corresponding sound buffer, say 410-1, when the buffer 410-1 is available. If the buffer 410-1 is full and/or otherwise unable to receive a new packet, the packet control module 508 sends the received packet to the overflow buffer 412 for subsequent processing.

The sound buffers A through N 410 provide sound data received from the packet control module 508 to the sound mixer 408, which is coupled to an output device such as a speaker for conversion to audible sound. In one embodiment of the invention, there are a maximum of four audio streams played simultaneously in a client computer. For each audio stream, there is a corresponding sound buffer in the client computer. Each audio stream may be thought of as representing a "phone line" between the client computer and a different user that is participating in the audio conference call with a user associated with the client computer. Although four audio streams are played simultaneously in one embodiment of the invention, any number of audio streams may be played simultaneously in accordance with the invention. However, playing more than four streams simultaneously may make the sound unintelligible and/or may exceed the data transmission bandwidth available to client computers, especially those connecting to the network at slower speeds.

The packet control module 508 shown in FIG. 5(B) may be implemented as a software routine or with the use of hardware circuits. There are various methods that can be used by the packet control module 508 to store and retrieve audio data in FIG. 5(B). In one embodiment of the invention, a FIFO (First In First Out) technique is used to determine the order in which incoming audio data is stored in and retrieved out of an overflow buffer (for example 412) of a client computer (for example 202). In alternate embodiments of the invention, however, any other suitable queuing system

BNSDOCID: <WO__0060809A1_I_>

may be used in conjunction with the present invention. For example, a priority-based queuing system, LIFO (Last In First Out), and other suitable queuing systems, alone or in combination, may be used in conjunction with the invention.

FIG. 6 illustrates functional components of the webtalk engine 406. The webtalk engine 406 comprises executable instructions forming a communication module 602, an audio module 604, and an interface module 606. The communication module 602 is configured to control communication between the client computer 202 and an associated audio server 240 or other computers.

The audio module 604 controls the packet control module 508, the sound buffer 410, and the sound mixer 408 to provide buffering, mixing, and related processing in connection with incoming and outgoing audio data packets. One example of the audio module 604 is described in U.S. Pat. No. 5,764,900 entitled "System and Method for Communicating Digitally-Encoded Acoustic Information Across a Network Between Computers," which is incorporated herein by reference. Those skilled in the art will recognize that the functions provided by the audio module 604 may be achieved in any number of alternative implementations. Finally, the interface module 606 is configured to control the flow of information between the user interface module 404 and the webtalk engine 406.

FIG. 7 illustrates a client computer (for example 202) and an audio server 240 connected through the network 206. In FIG. 7, the client computer 202 comprises the components shown in FIG. 3. The input/output devices 320 include a monitor 316 and a keyboard 718. In addition, the memory 304 of the client computer 202 comprises a contacts database 720 and a user profile database 722. Among other things, the contacts database 720 can be used to store the names of persons known to the user, persons with whom the user had a conference call previously, or other individuals having access to the telephony system. The user profile database 722 can be used to store information related to the user such as user name, password, and ID. The client computer 202 is coupled to the audio server 240 through the network 206.

The server 240 comprises a network interface 712, a CPU 714, and a server memory 700. A bus 716 interconnects the network interface 712, the CPU 714, and the server memory 700. The server memory 700 comprises a registered user database 702 for storing information about each user who is registered to log into the system

BNSDOCID: <WO___0060809A1_I_>

200. a webtalk server application 704 for controlling calls and multi-party conferences between or among client computers. and a region for storing user data 706. In addition. the server memory 700 may contain a buffer 708 and/or a conference participants database 710. In an alternate embodiment of the invention. the registered user database is maintained on a separate server to minimize the load on any particular computer.

The user data 706 will typically include. for each user connected to the audio server. the following information: a socket number associated with the connection. a nickname and/or user ID. a watch list (a list of users with whom the user frequently communicates). user status information (e.g.. whether the user is currently engaged in a call or conference. or otherwise unavailable to receive a call). and configuration parameters characterizing the nature of the connection of the user to the audio server (e.g.. whether the client computer associated with the user is located behind a firewall). The conference participant database 710 can be used to store the names or other identifiers of the participants of a conference call. The content of the conference participant database 710 changes as participants are added to or removed from a conference.

The buffer 708 of the audio server memory 700 is used to store overflow packets and prevent audio data loss in cooperation with the overflow buffer 412 of the client computer. It will be appreciated by one skilled in the art that the exact number and size of the buffer 708 of an audio server and the overflow buffer 412 of a client computer are not important to the invention as long as they provide adequate flow control to prevent audio data loss. For example. in one embodiment of the invention. the audio server may contain a large overflow buffer and the client computer a smaller overflow buffer while in an alternate embodiment of the invention. the audio server may contain a small overflow buffer and the client computer a larger overflow buffer.

The general architecture and processing associated with the invention has now been disclosed. Attention presently turns to a more detailed discussion of the system of the invention and the advantages associated with the disclosed technology. FIG. 8 illustrates a GUI of an audio communication application provided in accordance with one embodiment of the present invention. In particular. the figure shows a GUI 801 which includes a menu bar 802. a status symbol 805. a Do-Not-Disturb (DND) button

BNSDOCID: <WO__0060809A1_I_>

807. a Forum button 809. a contacts list 811. a user status indicator 813. a user list 815. a Time In Call indicator 817. a Mute button 819. a Text Chat button 821. a Hold/Pick Up button 823. a Hang Up button 825. a Add/Find users button 827. a Shadowtalk Launch button 829. and a caller status indicator 831.

Typically. a user on a client computer. such as one of the client computers shown in FIG. 2. activates the GUI of FIG. 8 by engaging an icon representing the audio communication application of the present invention. For example. the GUI shown in FIG. 8 is activated by engaging a "FireTalk" audio communication application icon on a client computer that includes the present invention.

The contacts list 811 is established by adding desired users to the list with the add button 827 or other suitable means. When the user status indicator 813 indicates "Ready." a user may start a group audio or text conference. When the user initiates a conference. the user status indicator 813 changes to "Active." A group audio conference is initiated by clicking on a user on the contacts list 811 who has a telephone symbol next to the user. which indicates that the user is on-line and available to receive a phone call. For example. BAT0 and Elise in the contacts list 811 are on-line and a group audio conference may be conducted among BAT0. Elise. and the current user.

When a user initiates a conference. participants in the conference are displayed in the user list 815. For example. in FIG. 8. a conference is in progress between the current user (CNF_Room) and Dave.

In accordance with the embodiment shown in FIG. 8. a user may receive a separate call while in a conference call by selecting the "Hold/Pick Up" button 823 on the main screen display of the audio communication application. Selecting the "Hold/Pick Up" button 823 places the user in communication with the incoming call and puts the original call/conference on hold. When the original call/conference is placed on hold. the caller status indicator 831 changes to "On Hold." A user may be in communication with multiple conference calls. While the user is communicating in a conference. other conferences may be placed on hold by selecting "Hold/Pick Up" button 823.

FIG. 9 illustrates processing steps for an audio communication application that may be executed in accordance with one embodiment of the invention. In the first

BNSDOCID: <WO___0060809A1_I_>

processing step shown in FIG. 9, the user interface module 404 of a client computer (for example 202) detects the user's call request to another user when the user uses a GUI (for example 801 of FIG. 8) of the present invention and clicks on an icon (*i.e.*, aligns a pointer with the person's icon and engages a mouse button) or on the text corresponding to the selected call recipient in the contacts list 811 (step 901). The interface module (for example 606) of the client computer then relays the information to the communication module (for example 602), which in turn relays the requested call to the server (for example 240) (step 903). The GUI of the user interface module 404 of the client computer 202 displays the name of the selected call recipient to the user window (for example the user list 815 in FIG. 8) of the client computer 202 (step 905). Thus, the list of conference participants is displayed in the user list 815.

The server reads the relayed information and identifies the call request to determine the source of the call request and the identity of the requesting party (step 907). The server notifies the selected call recipient of the call request and the identity of the requesting party (step 913). If the intended call recipient refuses to take the call (step 915), the server notifies the requesting party of the result (step 917). If the intended recipient accepts the call, the server registers the call recipient's name in its memory as a participant of a conference call and updates its memory, for example the conference participant database 710 of FIG. 7 (step 919). The server then establishes communication between the user who made the initial call request and the call recipient such that all audio data being sent from the user is forwarded to the client computer 202 corresponding to the call recipient (step 921), and all audio data from the call recipient is forwarded to the user who made the initial call request (step 923).

If a multi-party conference was already in progress in which the user is a participant, the server establishes communication in steps 921 and 923 such that each of the conference participants including the newly selected user are in audio communication with all other participants in the conference. Thus, while a conference is in progress, the server broadcasts the conference to all conference participants and each of the conference participants is in audio communication with each other.

At this point, unique attributes of the embodiment represented by FIGS. 8-9 will be apparent to those skilled in the art. The invention is highly advantageous because it provides a visual window where a user may see a list of contacts, select a

BNSDOCID: <WO__0060809A1_I_>

name therein, find a new contact, or initiate a conference in which all participants are in audio communication with each other. This functionality is achieved by combining a buffer control mechanism with a novel multi-party conferencing GUI, as will be described in greater detail below.

When the user wishes to find another user, the user can initiate a search by pressing the find button 827, which activates a "Find User" window. FIG. 10 illustrates an example of the "Find User" window. The "FindUser" window 1001 shown in FIG. 10 comprises "FireTalk ID" space 1003, "Nickname" space 1005, "Email" space 1007, "First Name" space 1009, "Last Name" space 1011, "Matching" list 1013, a "Find" button 1015, a "Add User" button 1017, a "Call" button 1019, and a "Cancel" button 1021.

FIG. 11 illustrates an example of a user option window 1101 that is activated when the user CNF_Room clicks the right mouse button after selecting the name BAT0 from the contacts list 811. If CNF_Room wants to block the user BAT0 from future conference calls, CNF_Room selects the "Block & Remove Contact" button from the user option window 1101. Each participant in a conference call may form a block list to exclude certain users from audio communication. In the example shown in Figure 8, CNF_Room may block other users from a group conference by setting appropriate parameters by, for example, engaging the right mouse button after selecting a user name from the contacts list 811.

FIG. 12 illustrates processing steps for creating a block list for a user. In FIG. 12, when a user selects a block menu from the user option window 1101 on a client computer (for example 202) and engages a mouse button to block another user, the GUI of the user interface module 404 detects the user's block request (step 1201). The interface module 606 of the webtalk engine 406 relays the block request received from the user interface module 404 to the communication module 602 (step 1203), which then transmits the information to a server (for example 240) (step 1205). The server, upon receiving the block request, identifies the user who is requesting the block and the user who is to be blocked (step 1207). The server then updates the user data 706 on its memory 700 to register the blocked user (step 1209), and stops all communication between the blocked user and the block-requesting user (step 1211).

BNSDOCID: <WO__0060809A1_I_>

Unlike prior art conference systems, the blocking mechanism of the present invention stores the blocking parameters set by the user in the client memory 304 in, for example, the user profile database 722. The blocking parameters may also be stored in the registered user database 702 in a server. In an alternate embodiment of the invention, the registered user database 702 containing the users' blocking parameters is stored in a separate off-line database so that even if the user logs on to the system from a different client computer, the user would not need to set the blocking parameters again because the blocking parameters may be retrieved from the off-line database for the user.

FIG. 13 illustrates a login window 1301 where a user is required to enter the user's ID and password before using the audio communication application of the present invention. A login window may be activated by selecting the "Login" button under "File" menu in menu bar 802. In FIG. 13, the login window 1301 comprises a "Create New Account" button 1302, a "FireTalk ID" space 1303, a password space 1305, a "Look Up FireTalk ID" button 1307, a "Forgot Password" button 1309, a "Login" button 1311, a "Cancel" button 1313, and a "Help" button 1315. A user selects the "Create New Account" button 1302 when the user wants to create a new audio communication application account. A user with an existing "FireTalk" account may enter the user's name and password in the "FireTalk ID" space 1303 and the password space 1305, respectively. The "Look Up FireTalk ID" button 1307 and the "Forgot Password" button 1309 are used to allow a user to find a user ID and password in case the user forgets them.

FIG. 14 illustrates processing steps for connecting to a server for a conference over the network. In FIG. 14, when a user selects a "Login" button under the "File" menu in the menu bar 803 on a client computer (for example 202), the GUI of the user interface module 404 detects the user's login request (step 1401). The interface module 606 relays the login request to the communication module 602 (step 1403), which then transmits the information to a server (for example 240) (step 1405). The server, upon receiving the login request, identifies the user who is requesting the login (1407), and verifies the identity of the login requester against a list of identifiers corresponding to authorized or registered users of the conference system 200 (step 1409).

BNSDOCID: <WO___0060809A1_I_>

The authentication step 1409 is an optional step. Further, in alternate embodiments of the invention, the authentication step 1409 may use more secure authentication procedures including handshaking and/or encryption technology.

If the login requester is not an authorized user, then the server sends an appropriate message to the client computer and disconnects the unauthorized user (step 1411). If the login requester is an authorized user, the server updates the user data 706 on its memory 700 to register the login user (step 1413), and continues communication with the client computer (step 1415).

FIG. 15 illustrates an example of a broadcast window 1501 opened by CNF_Room. CNF_Room initiates the broadcast window by selecting "Text Chat" button 821 on GUI 801, which opens the broadcast window 1501. When CNF_Room selects the "Text Chat" button 821, the client computer sends the selection to the server (for example 240), which then establishes a connection for text conferencing between CNF_Room's computer and the computer of the user with whom CNF_Room intends to conduct a text conference.

In the embodiment shown in FIG. 15, an audio conference and a text conference are conducted simultaneously between conference participants, as illustrated by user list 815 (FIG. 8) and the broadcast window 1501. When a text conference is conducted between participants of a conference call using the broadcast window, the text conference is visible to all the participants in the conference call, and all the participants see the same content in their broadcast window if they have a broadcast window open. In the embodiment shown in FIG. 15, a broadcast window for a text conference may be toggled between being displayed and not displayed by pressing the "Text Chat" button 821 (FIG. 8).

Although the broadcast window 1501 is used for text conferencing in the embodiment shown in FIG. 15, the broadcast window 1501 may also be used for other applications such as for creating, viewing, and editing documents, files, graphical images, video clips, and other electronic media.

In addition to conducting an audio conference and viewing a broadcast window, the invention enables conference participants to send peer-to-peer messages in accordance with the technology of the present invention. Participants in an audio or text conference and participants in peer-to-peer messaging need not be the same. For

BNSDOCID: <WO___0060809A1_I_>

example, while a conference is in progress between CNF_Room and Dave, a text message may be sent to CNF_Room by, for example, "Anthony" independent of the conference call between CNF_Room and Dave.

FIG. 16 illustrates a text messaging window 1601 activated in accordance with one embodiment of the invention. The text messaging window comprises a "FireTalk ID" space 1603 for entering FireTalk ID, a "Message" window 1605 for entering a text message, a "Cancel" button 1607 for canceling the messaging, and a "Send" button 1609 for sending the text message. In one embodiment of the invention, a user may activate a text messaging window by selecting a user and opening a user option window 801 shown in FIG. 8 by clicking on the right mouse button. The user then selects "Send Instant Message" button from the user option window 801 to open a text messaging window.

FIG. 17 illustrates processing steps performed by a server for transmission of audio data to client computers in accordance with one embodiment of the invention. In the first processing step of FIG. 17, the server (for example, 240) checks its buffer to see if there is some audio data that can be retrieved and transmitted (step 1701). If there is audio data to be sent in the buffer, the server checks to see if its transmission would cause a saturation condition at the destination client computer(s), which may result in an overflow condition and loss of audio data (step 1703). In one embodiment of the invention, the server monitors the volume of audio data sent to each client computer. If the volume of audio data transmission to a certain client computer reaches a predetermined value during a predetermined period of time, the server determines that further transmission to that particular client computer would saturate the client computer's receiving capacity.

If the server determines that a transmission would not cause a saturation condition, it retrieves audio data from the buffer (step 1705) and sends it to the destination client computer(s) (step 1707). If the server determines that a transmission would cause a saturation condition, the server then checks whether there is incoming audio data from a client computer (step 1715). If there is incoming audio data, the server places the received audio data in the buffer (step 1717).

If there is no audio data in the buffer in step 1701, the server then determines whether there is incoming audio data from a client computer (step 1709). If there is

BNSDOCID: <WO__0060809A1_I_>

incoming audio data. the server determines whether a transmission of audio data to a client computer would exceed the receiving capacity of that client computer or otherwise cause a saturation condition at the destination client computer(s) (step 1711). If the transmission is likely to cause a saturation condition. the server places the incoming audio data in the buffer of the server (step 1713). If it is safe to transmit. the server sends the audio data to the destination client computer(s) (step 1707).

FIG. 18 illustrates processing steps of a client computer for receiving audio data. The steps shown in FIG. 18 may be used by the packet control module 508 of FIG. 5B. In FIG. 18. the client computer (for example. 202) determines whether there is audio data in its overflow buffer (for example 412) (step 1801). If there is. the client computer 202 determines whether a sound buffer (for example 410) is available (step 1803). If there is an available sound buffer. the client computer retrieves the first audio data from its overflow buffer and feeds it to the available sound buffer (step 1805). If there is no audio data in its overflow buffer or there is no available sound buffer. the client computer proceeds to check whether there is any newly received audio data packet (step 1807).

If there is audio data packet newly received from other client computers. the client computer 202 determines whether a sound buffer is available (step 1809). If there is no available sound buffer. the client computer stores the newly received audio data in its overflow buffer (step 1811). If there is an available sound buffer. the client computer 202 feeds the newly received audio data to the available sound buffer (step 1813). The client computer's sound mixer (for example 408 of FIG. 4) then mixes the signals from the sound buffers (step 1815). and sends the mixed signals to an output device such as a speaker to convert the acoustic signal to audible sound (step 1817).

In a preferred embodiment of the invention. each sound buffer is allocated for a particular client computer so that audio data received from the same client computer is routed to the same buffer for the duration of speech according to the processing steps of FIG. 18. The end of the speech is indicated by the start/middle/end-of-data indicator contained in the payload 504 in one embodiment of the invention.

For example. the sound buffer 410-1 in FIG. 5(B) of the client computer 202 may be allocated to audio data received from the client computer 204. the sound buffer 410-2 to the client computer 210. and the sound buffer 410-3 to the client computer

BNSDOCID: <WO___0060809A1_I_>

212. Thus, when an audio data packet is received from the client computer 204 in step 1807, the client computer 202 checks the corresponding sound buffer 410-1 to determine whether a sound buffer is available for the client computer 204 in step 1809. If the sound buffer is available, audio data received from the client computer 204 is routed to the corresponding sound buffer 410-1 (step 1813). Otherwise, the audio data is routed to the overflow buffer 412 (step 1811). In this embodiment, each sound buffer receives audio data originated from a corresponding client computer in a proper sequence without interruption by audio data from other client computers for the duration of the speech.

Unique attributes of the invention will be recognizable to those skilled in the art at this point. Unlike prior art conferencing systems, the present invention, in accordance with the embodiments shown in FIGS. 17-18, provides an environment where users may engage in a conference where a plurality of users are allowed to speak simultaneously without losing information or audio data from other users. This functionality is achieved by combining a buffer control mechanism and a multi-party conference facility so that all audio data is eventually delivered to the destination computers, although some audio data may be delayed.

FIG. 19 illustrates an alternate embodiment of processing steps for receiving audio data for a client computer. The steps shown in FIG. 19 may be used by the packet control module 508 of FIG. 5B. In FIG. 19, the steps in FIG. 19 generally correspond to the steps shown in FIG. 18 with one notable exception. Unlike the embodiments shown in FIGS. 17 and 18, the embodiment shown in FIG. 19 eliminates the need for a client computer to rely on an audio server to control audio data transmission.

In the embodiment shown in FIG. 19, the client computer determines whether an overflow buffer is available on the client computer (step 1917). If there is an available overflow buffer, the audio data is placed in the overflow buffer according to a suitable queuing method such as FIFO (step 1919). The client computer then checks the status of its overflow buffer and determines whether it reached its capacity (step 1923). In one embodiment of the invention, the client computer (for example, 202) compares the volume of received audio data with the available space of its buffers 410 and 412. When the volume of audio data equals a first predetermined percentage of

BNSDOCID: <WO__0060809A1_I_>

the available buffer space, the client computer considers the buffers as having reached their capacity. The client computer, if the buffer has reached its capacity or there is no available space in overflow buffer, transmits an "off " signal to all computers participating in the conference call, instructing the computers to stop transmission until further notice, and changes its reception state to "off" (step 1921).

When space becomes available in the overflow buffer (step 1905), the client computer sends an "on" signal to the computers in the conference call and changes the reception state to "on" (step 1909). The computers in the conference call, in response to the "on" signal, reinitiate transmission of audio data (including the stored data) to the client computer 202.

Unique attributes of the embodiment shown FIG. 19 will be recognizable to those skilled in the art. Unlike prior art conferencing systems, the present invention, in accordance with the embodiment shown in FIG. 19, provides an environment where users may engage in a conference where a plurality of users are allowed to speak simultaneously without losing information or audio data from other users without requiring a server's assistance in buffer management. This highly desirable functionality is achieved by implementing a buffer control mechanism on the client computers and combining it with a multi-party conferencing system.

Figure 20 illustrates an alternate embodiment for processing steps for controlling audio data transmission for a client computer. The steps shown in FIG. 20 may be used by the packet control module 508 of FIG. 5B. As with the embodiment shown in FIG. 19, the embodiment shown in FIG. 20 provides an audio data transmission control method that enables the client computer to control audio data transmission and avoid overflow conditions without relying on an audio server.

In FIG. 20, the client computer (for example 202) monitors the volume of incoming audio data and compares this volume with the available bandwidth. If the volume of incoming audio data reaches a first predetermined percentage ("upper level") of the maximum bandwidth of the client computer (step 2001), the client computer determines that its bandwidth or reception capacity has been saturated. If the state of data reception for the client computer is "on" (step 2003), the client computer transmits an "off " signal to all computers participating in the conference

BNSDOCID: <WO__0060809A1_I_>

call. instructing the computers to stop transmission until further notice. and changes the state of data reception to "off" (step 2005).

If the volume of incoming audio data is below the upper level. the client computer determines whether the volume of incoming audio data is at or below a second predetermined percentage ("lower level") of the maximum bandwidth (step 2007). If the volume of incoming audio data is at or below the lower level. the client computer determines the state of data reception (step 2009). If the state of data reception is "off." the client computer transmits an "on " signal to all computers participating in the conference call. instructing the computers to resume transmission. and changes the state of data reception to "on" (step 2011).

As an illustration of the upper level. if the client computer 202 is connected to the network 206 through a modem having a 3.200 bytes/sec reception capacity. the upper level may be predetermined to be 75%. Thus. in this example. when the volume of incoming audio data reaches 75% of 3.200 bytes/sec or 2.400 bytes/sec. the client computer transmits an "off" signal. The exact figure of the upper level is not important to the invention and may vary depending on the needs of a particular application. For example. if the client computer is connected to the network 206 via a faster interface such as a LAN or a T-1 interface. the upper level can be in a range between 500 Kbytes/sec and 800 Kbytes/sec. Likewise. the lower level may be set at 25% of the maximum bandwidth in one embodiment of the invention. However. the exact figure of the lower level is not important. and may vary depending on a particular application.

Unique attributes of the embodiment shown FIG. 20 will be recognizable to those skilled in the art. In addition to the unique attributes achieved by the embodiment of FIG. 19. the embodiment shown in FIG. 20. eliminates the need to rely on buffers to control overflow conditions. This highly desirable functionality is achieved by monitoring the available bandwidth of a client computer to detect a saturation level. and sending an appropriate signals to other computers to avoid overflow conditions.

It will be appreciated by one skilled in the art that the processing steps shown in FIG. 20 may be implemented separately or combined with the processing steps shown in FIG 18 in order to provide audio data transmission control and prevent

BNSDOCID: <WO__0060809A1_I_>

bandwidth saturation conditions. In a preferred embodiment of the invention, the steps shown in FIG. 20 are implemented separately from the steps shown in FIG. 18, and are executed periodically (for example, every second) by the client computer.

FIG. 21 illustrates processing steps for conducting multiple calls and conferences for a client computer (for example 202) in accordance with one embodiment of the invention. The first step shown in FIG. 21 is for the client computer to determine whether a conversation request to communicate with the client computer is being made by another client computer (step 2101). If there is a conversation request, the client computer determines whether the user selects "Hold/Pick Up" button 823 and accepts the conversation request (step 2103). If the user does not accept, the client computer determines whether the user selects the "Hang Up" button 825 (step 2105). If the user selects the "Hang Up" button 825, the client computer disconnects the incoming call (step 2109)

If the user decides to accept the conversation request, the client computer checks to see if any previously initiated conversation is still in progress (step 2111). If there is one, the client computer places the current ongoing conversation on hold (step 2113) and initiates communication with the new incoming caller (step 2115).

FIG. 22 illustrates processing steps of a server for establishing a peer-to-peer messaging session in accordance with one embodiment of the invention. The first processing step of FIG. 22 is for a server (for example 240) to determine whether peer-to-peer messaging is requested by a client computer (step 2201). If there is a peer-to-peer messaging request, the server establishes communication with the recipient client computer indicated by the requesting client computer (step 2203).

Next, the server checks to see if any client computer is behind a firewall (step 2205). If neither client computer is behind a firewall, the server sends instructions to both client computers to establish direct peer-to-peer communication (step 2207). If, however, either or both of the client computers are behind a firewall, direct peer-to-peer messaging is not available and the server sends instructions to both client computers to establish communication via the server (step 2209). Direct peer-to-peer audio messaging is available when, for example, both client computers are capable of using UDP as their communication protocol.

The foregoing descriptions of specific embodiments of the present invention are presented for purposes of illustration and description. They are not intended to be exhaustive or to limit the invention to the precise forms disclosed. obviously many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles of the invention and its practical applications. to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalents.

BNSDOCID: <WO__0060809A1_I_>

WHAT IS CLAIMED IS:

1. A method of inviting candidate users to a conference call comprising one or more participants, wherein each participant is associated with a different end-user computer, the method comprising:

originating a conference call request from a participant by selecting a candidate user from a list of candidate users that is displayed on a display screen of an originating end-user computer associated with said participant;

transmitting said conference call request from said originating end-user computer to a selected end-user computer associated with said selected candidate user;

notifying said selected candidate user of said conference call request and the identity of said participant originating said conference call request when said selected end-user computer receives said call request;

joining said selected end-user computer to said conference call when said candidate user accepts said conference call request, wherein said candidate user becomes a first additional participant in said conference call;

indicating on a display screen of each end-user computer associated with a participant in said conference call that said first additional participant has joined said conference call; and

changing a status symbol associated with said first additional participant on said display screen of said selected end-user computer to indicate that said first additional participant is now in said conference call.

2. The method of claim 1 wherein said originating step includes the step of displaying a list of candidate users from a personal contact list stored on said originating end-user computer.

3. A method of communicating between a first end-user computer associated with a first user, and one or more second computers, each of which is associated with a user, wherein said first and said one or more second end-user computers each has a display screen and is in communication with a server, said method comprising the steps of:

BNSDOCID: <WO__0060809A1_I_>

forwarding a first signal from said first end-user computer to said server when said first user speaks, said first signal including a digital representation of a speech;

distributing said first signal from said server to each of said one or more second end-user computers;

receiving, at said server, one or more responsive signals from said one or more second end-user computers, each of said one or more responsive signals generated by a different second end-user computer when a user associated with said different second end-user computer speaks, each of said one or more responsive signals including a digital representation of said speech;

directing an initial responsive signal of said one or more responsive signals from said server to said first end-user computer;

storing subsequent responsive signals of said one or more responsive signals in a buffer; and

sequentially converting said subsequent responsive signals from said buffer to sound at said first end-user computer.

4. The method of claim 3 wherein said storing step is performed by said first end-user computer.

5. The method of claim 3 wherein said storing step is performed by said server.

6. The method of claim 4 further comprising the steps of:

sending an off signal to said one or more second end-user computers when a bandwidth of said first end-user computer reaches a first predetermined level, thereby causing said one or more second end-user computers to stop transmission to said first end-user computer; and

sending an on signal to said one or more second end-user computers when said bandwidth of said first end-user computer reaches a second predetermined level, thereby causing said one or more second end-user computers to resume transmission to said first end-user computer.

BNSDOCID: <WO__0060809A1_I_>

7. The method of claim 3 further comprising the step of providing the first user with the option to set a pause option: wherein, when said pause option is set,

each of said one or more responsive signals is allocated to a corresponding buffer selected from a plurality of buffers associated with said first end-user computer while said first user is speaking: and

a subset of said one or more responsive signals and said plurality of buffers is converted to sound when said first user is not speaking.

8. The method of claim 7 wherein said subset is limited to a maximum of four signals at any given time.

9. The method of claim 3 further comprising the step of

adding a digital representation of an identity of said first user to said first signal.

10. The method of claim 3 further comprising the steps of:

displaying a broadcast window on said display screen of said first end-user computer;

creating a broadcast window message when a message is typed into said broadcast window of said first end-user computer:

sending said broadcast message from said first end-user computer to said server:

distributing said broadcast message from said server to said one or more second end-user computers: and

updating said broadcast window on each of said one or more second end-user computers with said broadcast message.

11. The method of claim 3 wherein said one or more responsive signals are rejected by said first end-user computer if said one or more responsive signals are from a user that is on a block list of said first user.

BNSDOCID: <WO___0060809A1_I_>

12. The method of claim 3 further comprising the steps of:

opening a message window on said display screen of said first end-user computer when said first user selects an indicator that represents a displayed identity of a second user that is associated with one of said one or more second end-user computers:

transmitting said message from said first end-user computer to said second end-user computer associated with said second user: and

displaying a message window on said display screen of said second end-user computer. said message window including a text of said message from said first user.

13. The method of claim 12 further including the step of providing said second user with an option to set a do-not-disturb toggle. and when said do-not-disturb toggle is set on. said message window is not displayed on said display screen of said second end-user computer

14. The method of claim 12 wherein said message window is not displayed on said display screen of said second end-user computer when said first user is on a block list of said second user.

15. A method of responding to a conversation request sent to a first user from a second user associated with a second computer: wherein said first user is associated with a first computer having a display screen. said method comprising:

displaying a conversation request message on said display screen of said first computer that identifies said second user while said first user is engaged in one or more conversations with other users:

providing said first user with an option to accept said conversation request; wherein. when said first user accepts said conversation request message. said first user is connected with said second user: and

updating said display screen of said first computer: wherein said display screen identifies each conversation in which said first user is participating and the update indicates that said first user is speaking with said second user and that conversations with other users have been placed on hold.

BNSDOCID: <WO__0060809A1_I_>

16. The method of claim 15 further comprising the concurrent step of providing said first user with an option to set a do not disturb toggle, and when said do not disturb toggle is set on, said conversation request message is rejected and is not displayed on said display screen of said first computer

17. The method of claim 15 wherein said conversation request message is not displayed on said display screen of said first computer when said second user is on a block list of said first user.

BNSDOCID: <WO___0060809A1_I_>

AMENDED CLAIMS

[received by the International Bureau on 25 September 2000 (25.09.00);
original claims 1, 7, 15 and 16 amended;
remaining claims unchanged (4 pages)]

1. A method of inviting candidate users to a conference call comprising one or more participants, wherein each participant is associated with a different end-user computer, the method comprising:

originating a conference call request from a participant by selecting a candidate user from a list of candidate users that is displayed on a display screen of an originating end-user computer associated with said participant;

transmitting said conference call request from said originating end-user computer to a selected end-user computer associated with said selected candidate user;

notifying said selected candidate user of said conference call request and an identity of said participant originating said conference call request when said selected end-user computer receives said call request;

joining said selected candidate user to said conference call when said selected candidate user accepts said conference call request, wherein said selected candidate user becomes an additional participant in said conference call;

indicating on a display screen of each end-user computer associated with a participant in said conference call that said additional participant has joined said conference call; and

changing a status symbol associated with said additional participant on said display screen of said selected end-user computer to indicate that said additional participant is now in said conference call.

2. The method of claim 1 wherein said originating step includes the step of displaying a list of candidate users from a personal contact list stored on said originating end-user computer.

3. A method of communicating between a first end-user computer associated with a first user, and one or more second computers, each of which is associated with a user, wherein said first and said one or more second end-user computers each has a display screen and is in communication with a server, said method comprising the steps of:

BNSDOCID. <WO__0060809A1_I_>

7. The method of claim 3 further comprising the step of providing said first user with the option to set a pause option; wherein, when said pause option is set,

each of said one or more responsive signals is allocated to a corresponding buffer selected from a plurality of buffers associated with said first end-user computer while said first user is speaking; and

a subset of said one or more responsive signals allocated to said plurality of buffers is converted to sound when said first user is not speaking.

8. The method of claim 7 wherein said subset is limited to a maximum of four signals at any given time.

9. The method of claim 3 further comprising the step of

adding a digital representation of an identity of said first user to said first signal.

10. The method of claim 3 further comprising the steps of:

displaying a broadcast window on said display screen of said first end-user computer;

creating a broadcast window message when a message is typed into said broadcast window of said first end-user computer;

sending said broadcast message from said first end-user computer to said server;

distributing said broadcast message from said server to said one or more second end-user computers; and

updating said broadcast window on each of said one or more second end-user computers with said broadcast message.

11. The method of claim 3 wherein said one or more responsive signals are rejected by said first end-user computer if said one or more responsive signals are from a user that is on a block list of said first user.

BNSDOCID: <WO___0060809A1_I_>

12. The method of claim 3 further comprising the steps of:

opening a message window on said display screen of said first end-user computer when said first user selects an indicator that represents a displayed identity of a second user that is associated with one of said one or more second end-user computers;

transmitting said message from said first end-user computer to said second end-user computer associated with said second user; and

displaying a message window on said display screen of said second end-user computer, said message window including a text of said message from said first user.

13. The method of claim 12 further including the step of providing said second user with an option to set a do-not-disturb toggle, and when said do-not-disturb toggle is set on, said message window is not displayed on said display screen of said second end-user computer.

14. The method of claim 12 wherein said message window is not displayed on said display screen of said second end-user computer when said first user is on a block list of said second user.

15. A method of responding to a conversation request sent to a first user from a second user associated with a second computer; wherein said first user is associated with a first computer, said first and second computers having a display screen, said method comprising:

displaying a conversation request message on said display screen of said first computer that identifies said second user while said first user is engaged in one or more conversations with other users;

providing said first user with an option to accept said conversation request; wherein, when said first user accepts said conversation request message, said first user is connected with said second user; and

updating said display screen of said first computer; wherein said display screen identifies each conversation in which said first user is participating and the update indicates that said first user is speaking with said second user and that conversations with other users have been placed on hold.

BNSDOCID: <WO__0060809A1_I_>

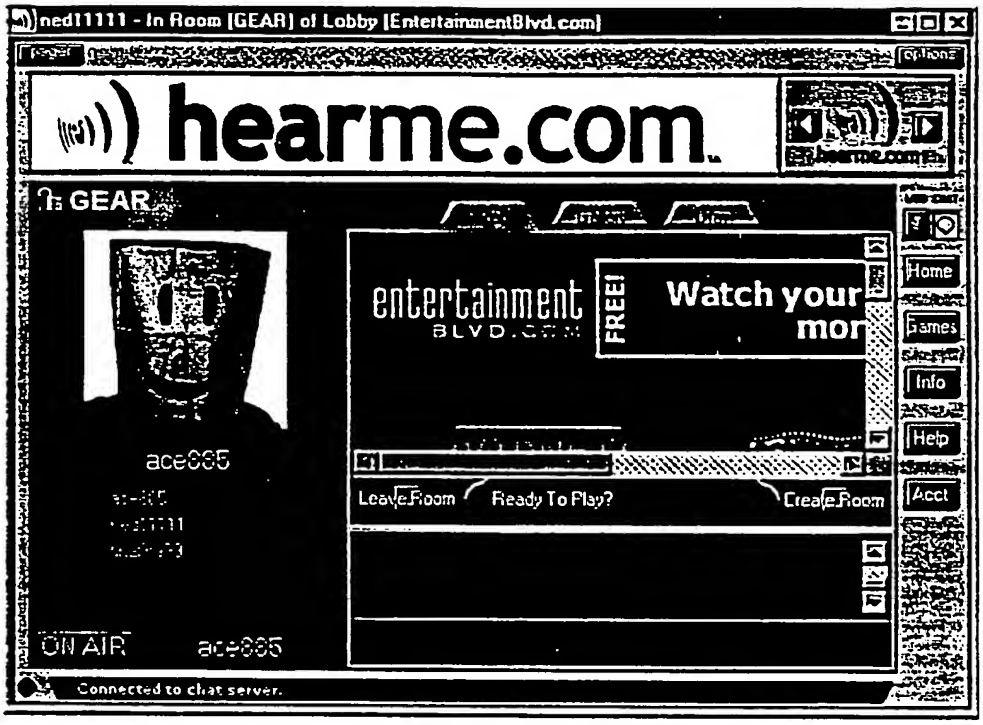


Figure 1
(Prior Art)

16. The method of claim 15 further comprising the concurrent step of providing said first user with an option to set a do-not-disturb toggle, and when said do-not-disturb toggle is set on, said conversation request message is rejected and is not displayed on said display screen of said first computer.

17. The method of claim 15 wherein said conversation request message is not displayed on said display screen of said first computer when said second user is on a block list of said first user.

BNSDOCID: <WO___0060809A1_I_>

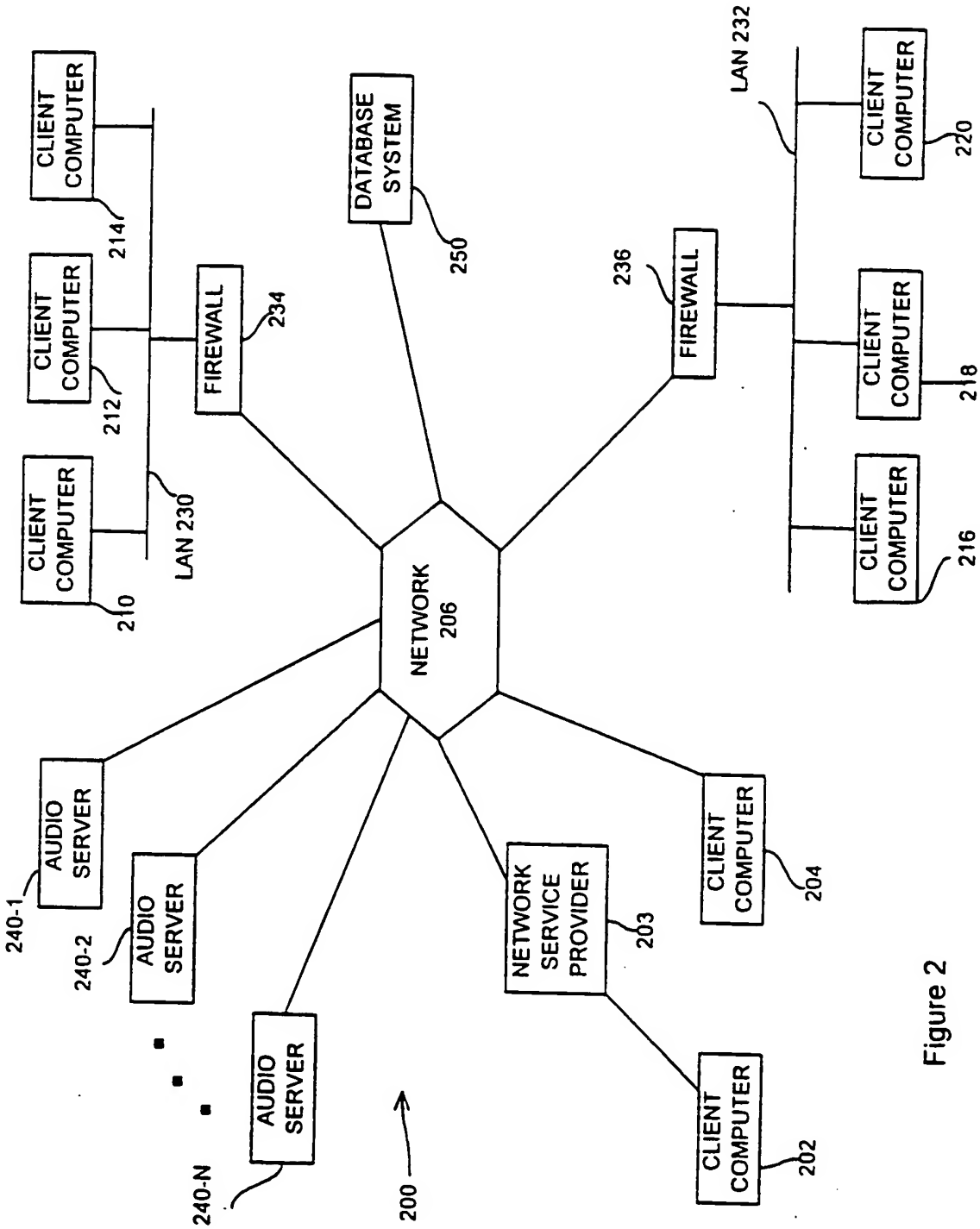

CLIENT COMPUTER
214
CLIENT COMPUTER
212
CLIENT COMPUTER
210
FIREWALL
234
LAN 230
DATABASE SYSTEM
250
NETWORK 206
FIREWALL
236
LAN 232
CLIENT COMPUTER
220
CLIENT COMPUTER
218
CLIENT COMPUTER
216
AUDIO SERVER
240-1
AUDIO SERVER
240-2
AUDIO SERVER
240-N
NETWORK SERVICE PROVIDER
203
CLIENT COMPUTER
204
CLIENT COMPUTER
202
200

Figure 2

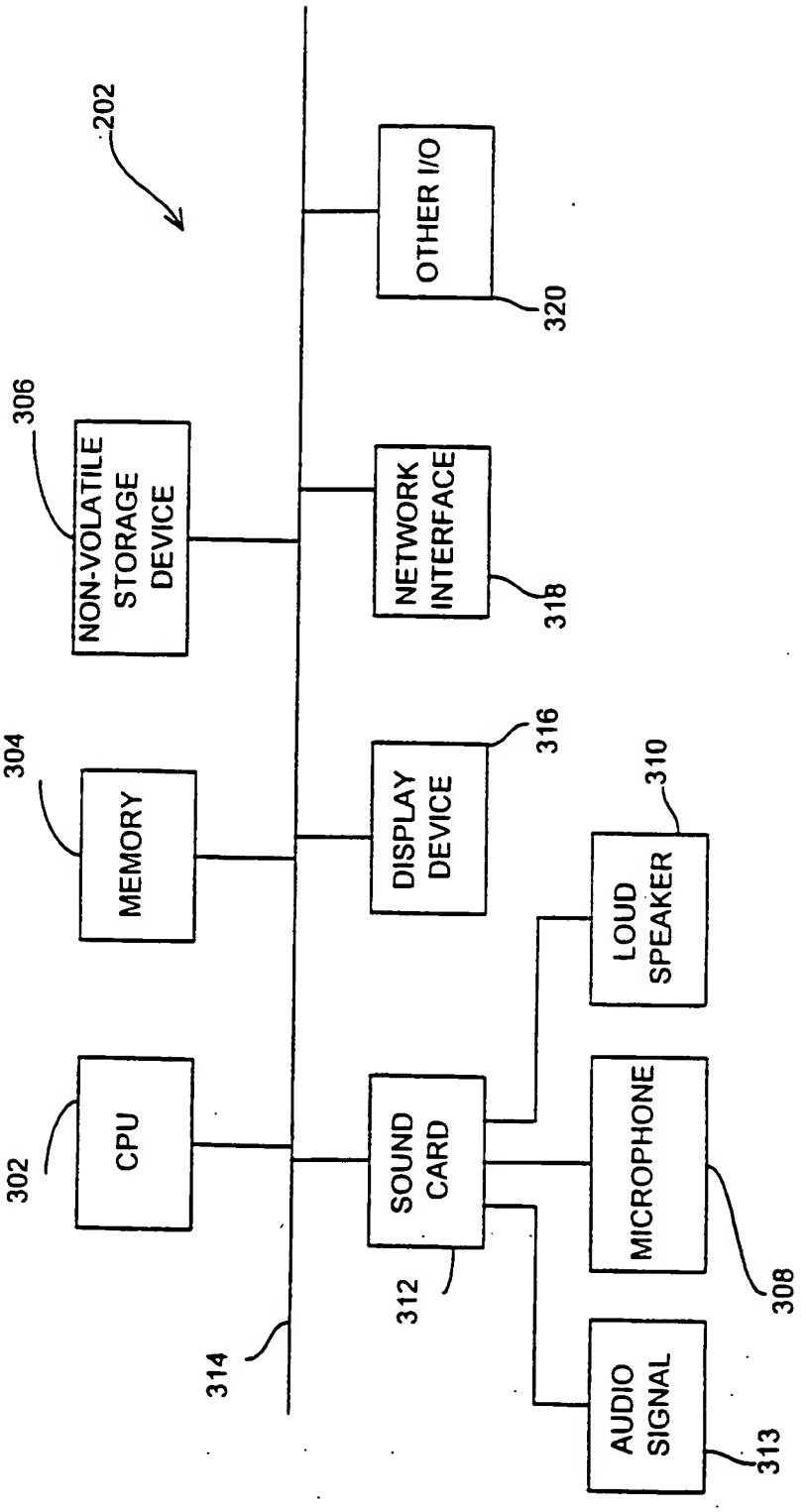

202
OTHER I/O
320
306
NON-VOLATILE STORAGE DEVICE
NETWORK INTERFACE
318
304
MEMORY
DISPLAY DEVICE
316
310
LOUD SPEAKER
302
CPU
SOUND CARD
MICROPHONE
312
308
314
AUDIO SIGNAL
313

Figure 3

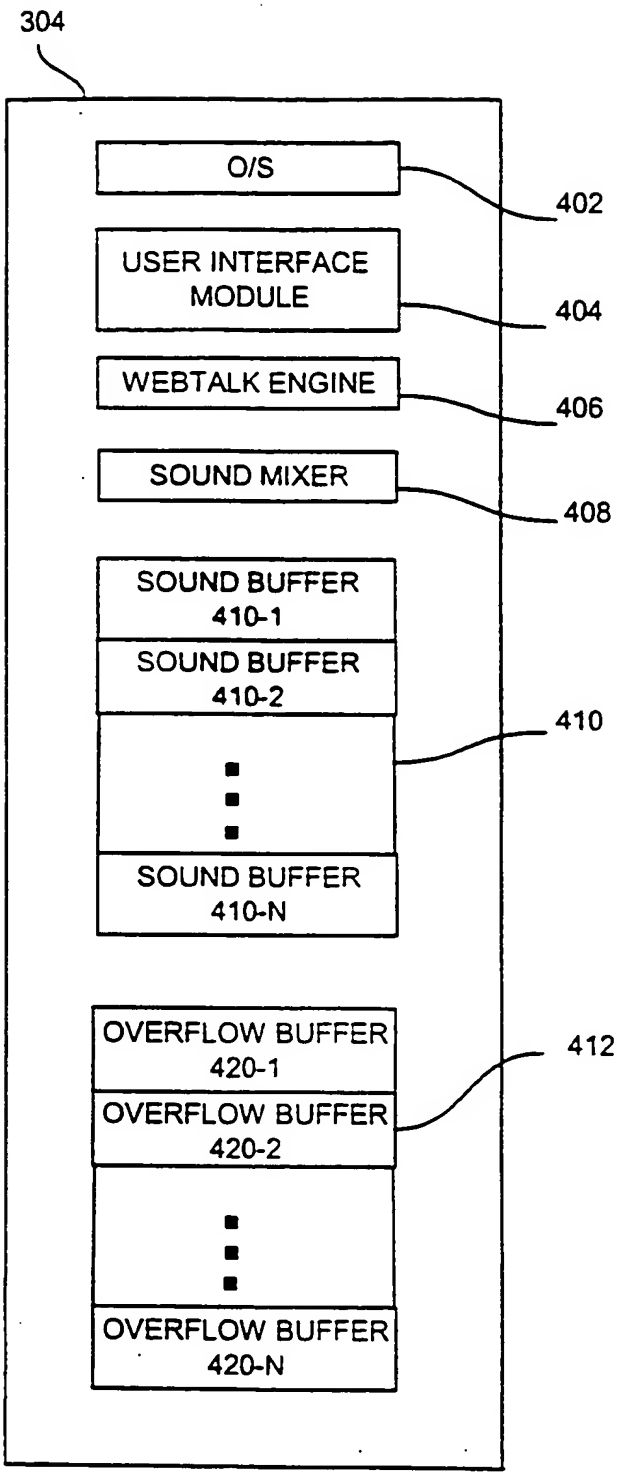

304
O/S
402
USER INTERFACE MODULE
404
WEBTALK ENGINE
406
SOUND MIXER
408
SOUND BUFFER 410-1
SOUND BUFFER 410-2
SOUND BUFFER 410-N
410
OVERFLOW BUFFER 420-1
OVERFLOW BUFFER 420-2
OVERFLOW BUFFER 420-N
412

Figure 4

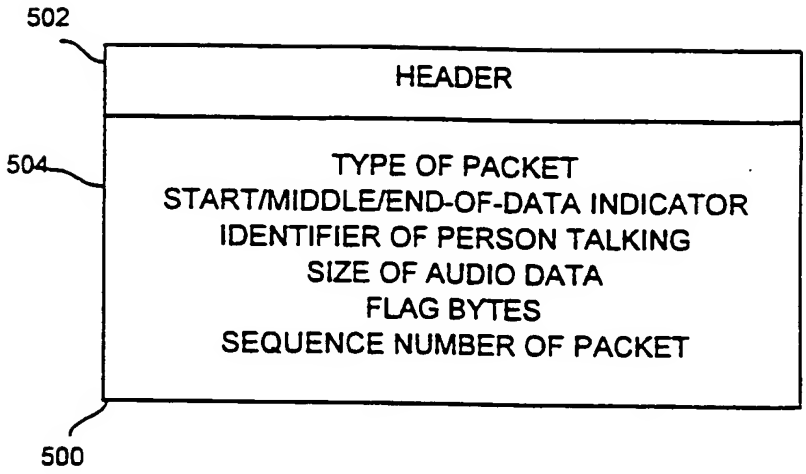

502
HEADER
504
TYPE OF PACKET
START/MIDDLE/END-OF-DATA INDICATOR
IDENTIFIER OF PERSON TALKING
SIZE OF AUDIO DATA
FLAG BYTES
SEQUENCE NUMBER OF PACKET
500

Figure 5(A)

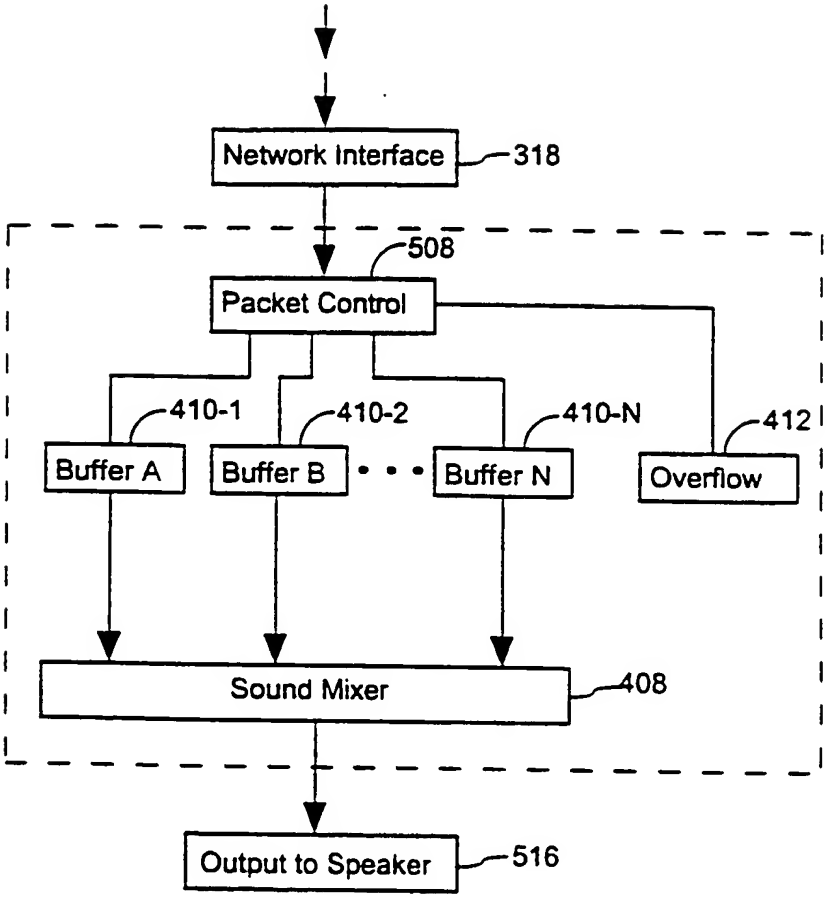

Network Interface
318
508
Packet Control
410-1
410-2
410-N
412
Buffer A
Buffer B
Buffer N
Overflow
Sound Mixer
408
Output to Speaker
516

Figure 5(B)

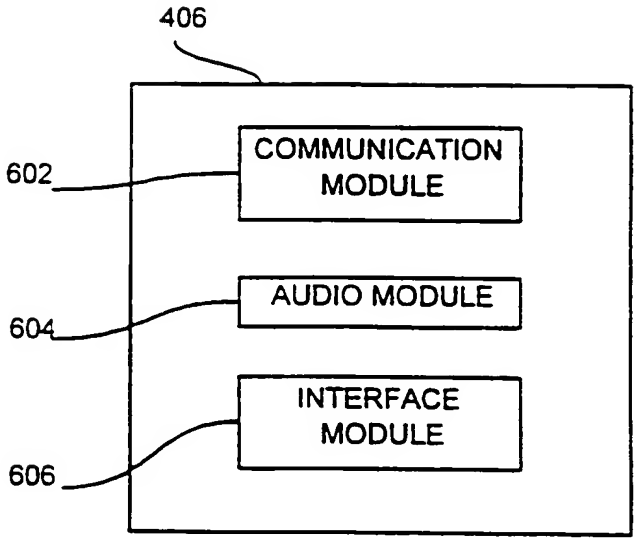

406
602
COMMUNICATION MODULE
604
AUDIO MODULE
606
INTERFACE MODULE

Figure 6

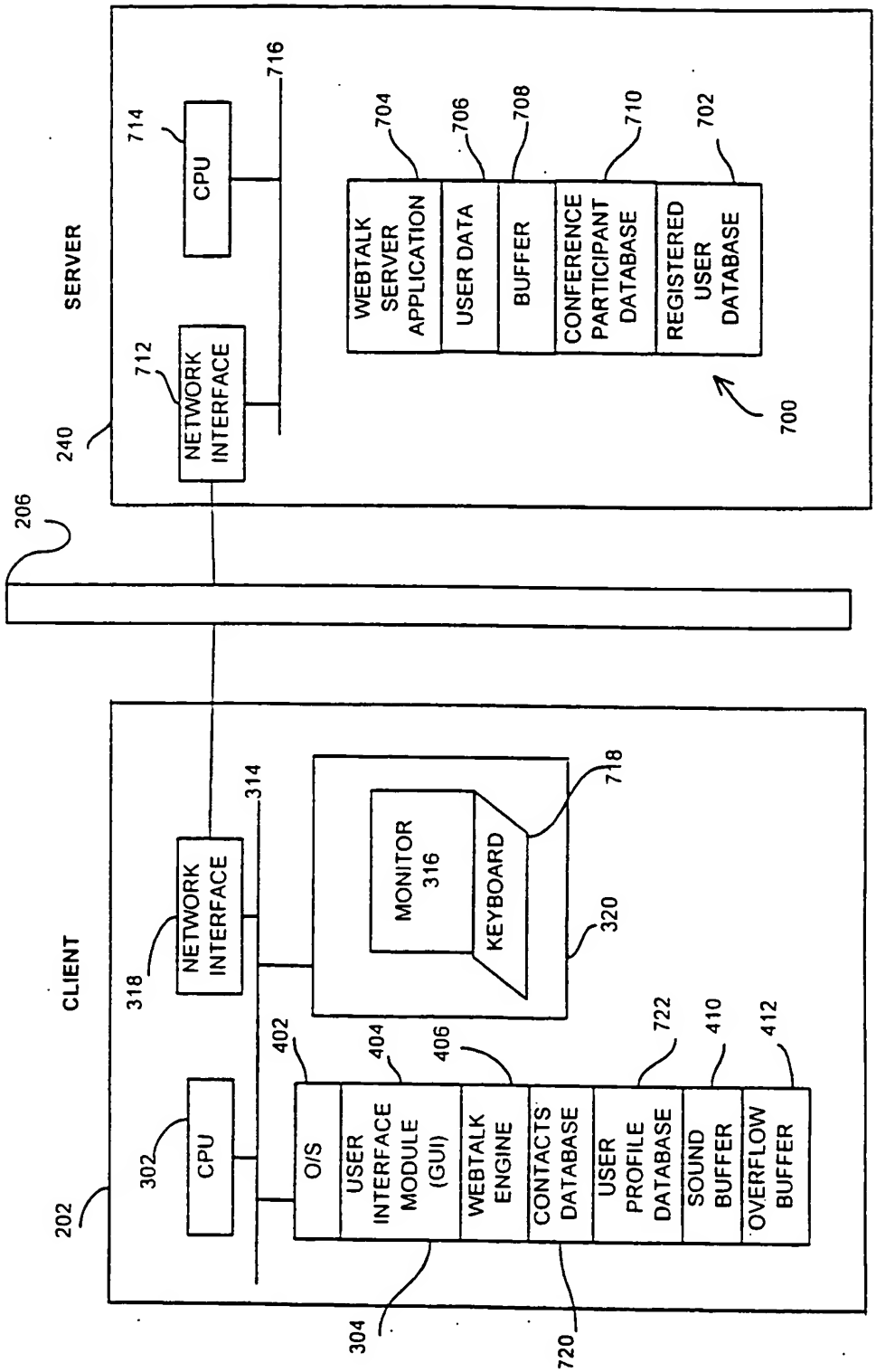

SERVER
240
206
CLIENT
202
CPU
302
NETWORK INTERFACE
318
314
NETWORK INTERFACE
712
CPU
714
716
MONITOR 316
KEYBOARD
718
320
O/S
402
USER INTERFACE MODULE (GUI)
404
WEBTALK ENGINE
406
CONTACTS DATABASE
720
USER PROFILE DATABASE
722
SOUND BUFFER
410
OVERFLOW BUFFER
412
304
WEBTALK SERVER APPLICATION
704
USER DATA
706
BUFFER
708
CONFERENCE PARTICIPANT DATABASE
710
REGISTERED USER DATABASE
702
700

Figure 7

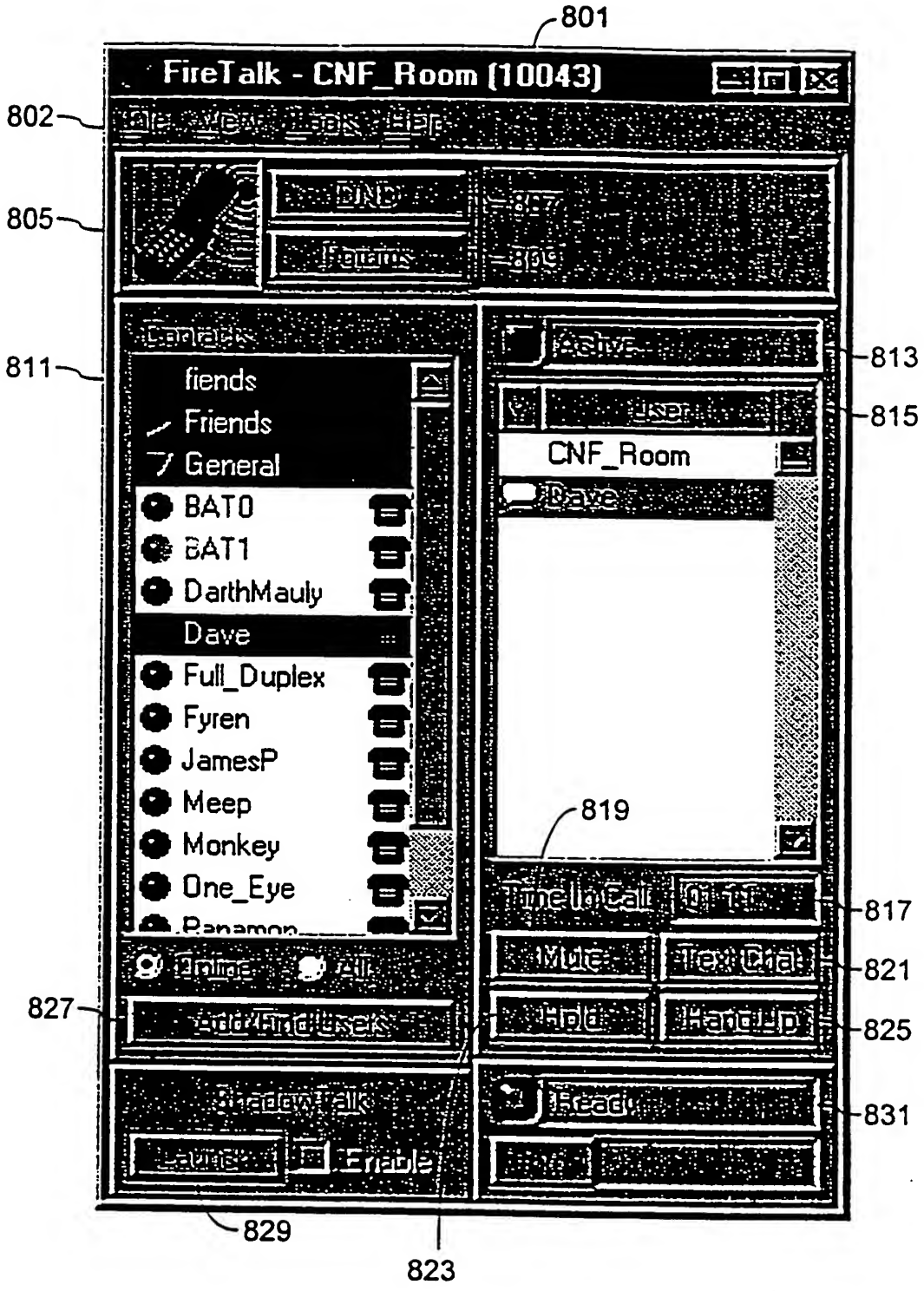

801
FireTalk - CNF_Room (10043)
802
File View Tools Help
805
DND
Forums
811
Contacts
fiends
Friends
General
BAT0
BAT1
DarthMauly
Dave
Full_Duplex
Fyren
JamesP
Meep
Monkey
One_Eye
Panamon
Online
All
Add/Find Users
ShadowTalk
Enable
813
Active
815
User
CNF_Room
Dave
819
Time In Call
817
Mute
Text Chat
821
Hold
Hang Up
825
Ready
831
827
829
823

Figure 8

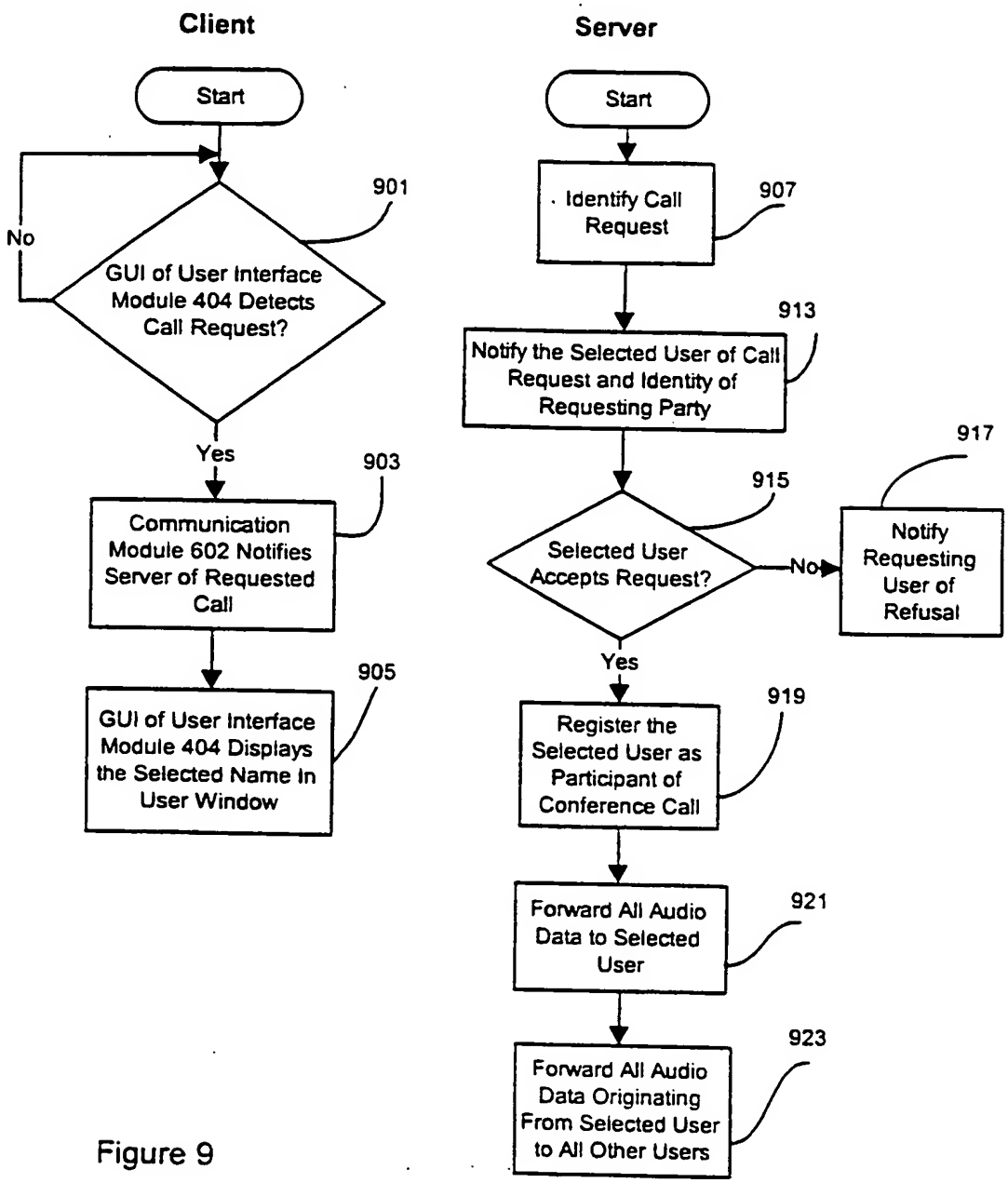

Client
Server
Start
Start
901
No
GUI of User Interface Module 404 Detects Call Request?
Yes
903
Communication Module 602 Notifies Server of Requested Call
905
GUI of User Interface Module 404 Displays the Selected Name in User Window
907
Identify Call Request
913
Notify the Selected User of Call Request and Identity of Requesting Party
915
Selected User Accepts Request?
No
917
Notify Requesting User of Refusal
Yes
919
Register the Selected User as Participant of Conference Call
921
Forward All Audio Data to Selected User
923
Forward All Audio Data Originating From Selected User to All Other Users

Figure 9

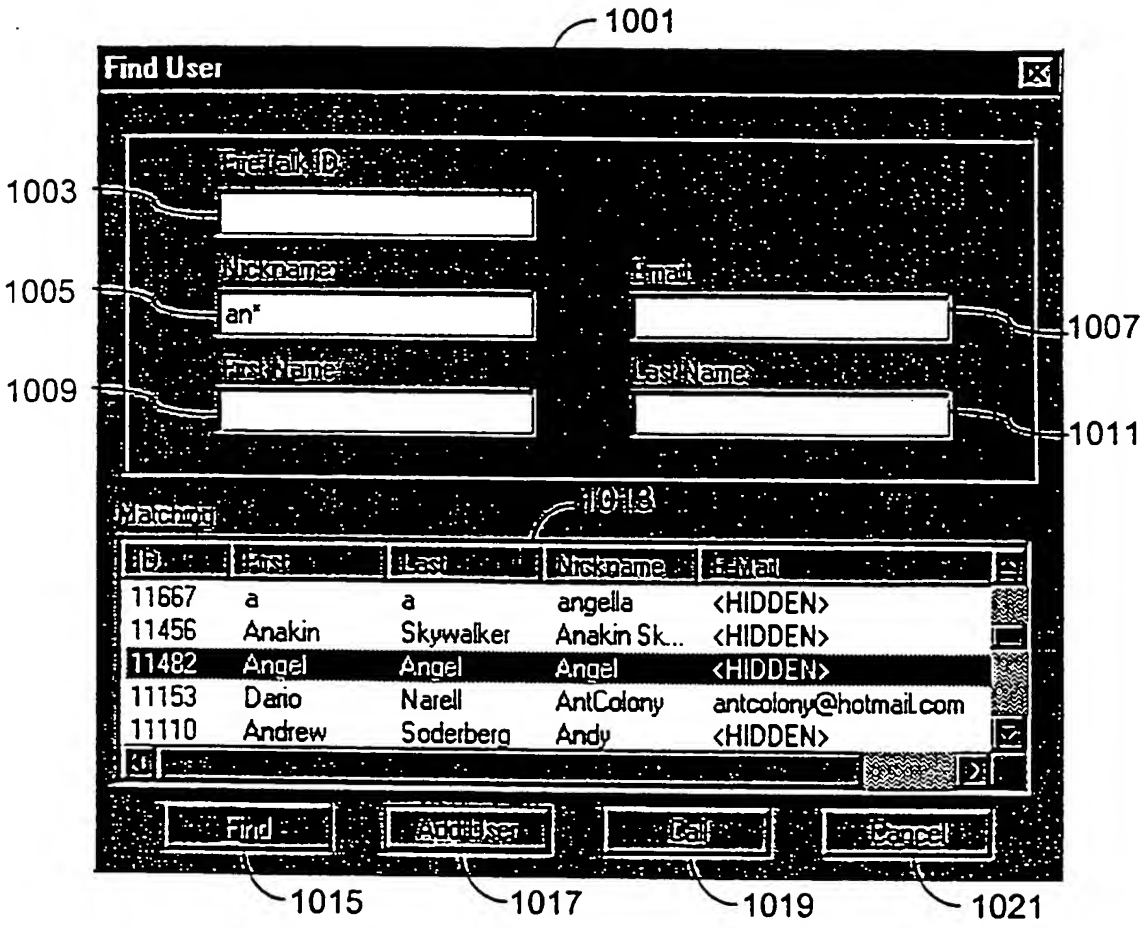

1001
Find User
FireTalk ID
1003
Nickname
1005
an*
Email
1007
First Name
1009
Last Name
1011
Matching
1013
ID
First
Last
Nickname
EMail
11667 a a angella <HIDDEN>
11456 Anakin Skywalker Anakin Sk... <HIDDEN>
11482 Angel Angel Angel <HIDDEN>
11153 Dario Narell AntColony antcolony@hotmail.com
11110 Andrew Soderberg Andy <HIDDEN>
Find
Add User
Edit
Cancel
1015
1017
1019
1021

Figure 10

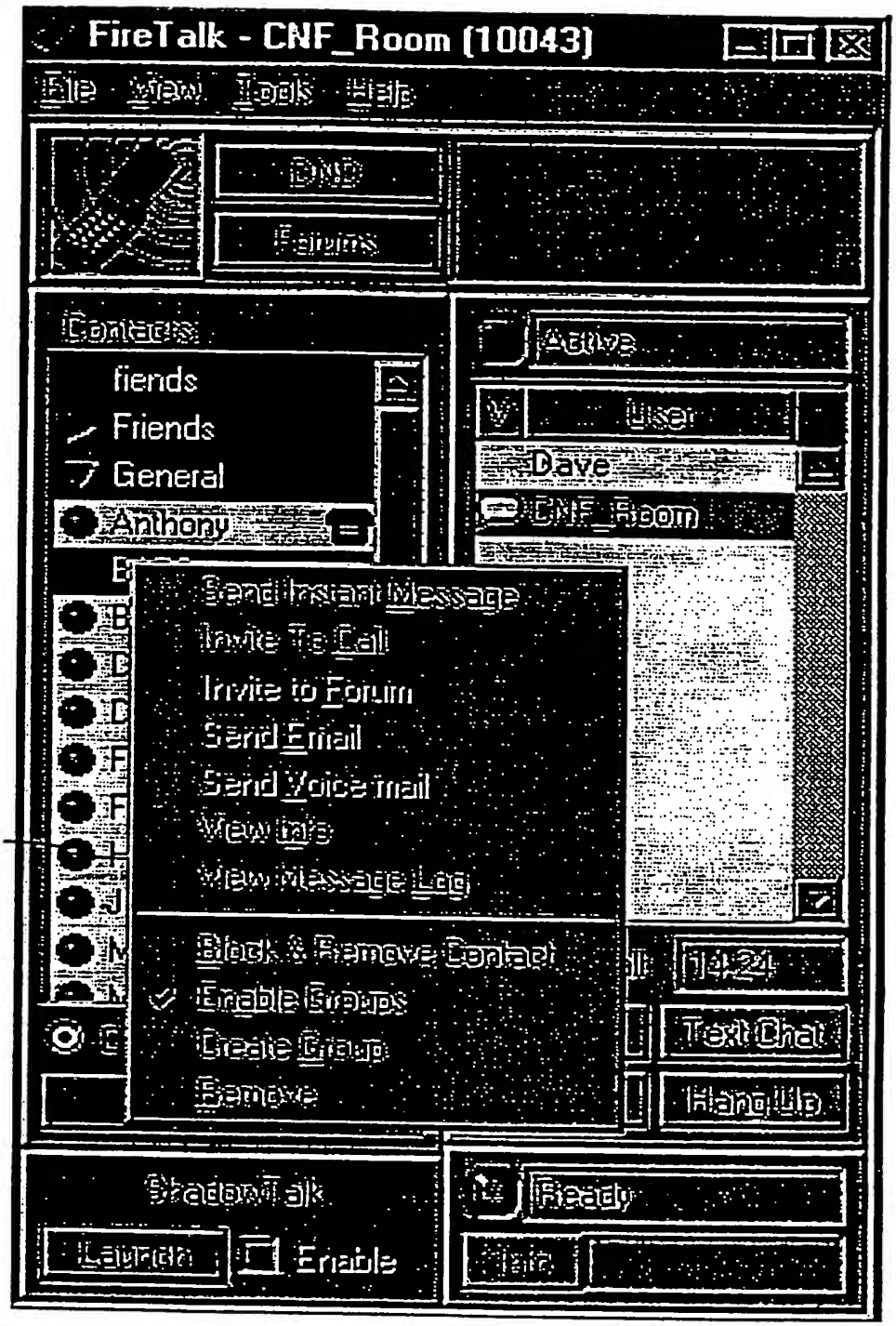

FireTalk - CNF_Room (10043)
File View Tools Help
DND
Forums
Contacts
fiends
Friends
General
Anthony
Active
User
Dave
CNF_Room
Send Instant Message
Invite To Call
Invite to Forum
Send Email
Send Voice mail
View Info
View Message Log
Block & Remove Contact
Enable Groups
Create Group
Remove
Text Chat
Hang Up
Ready
Launch
Enable
1101

Figure 11

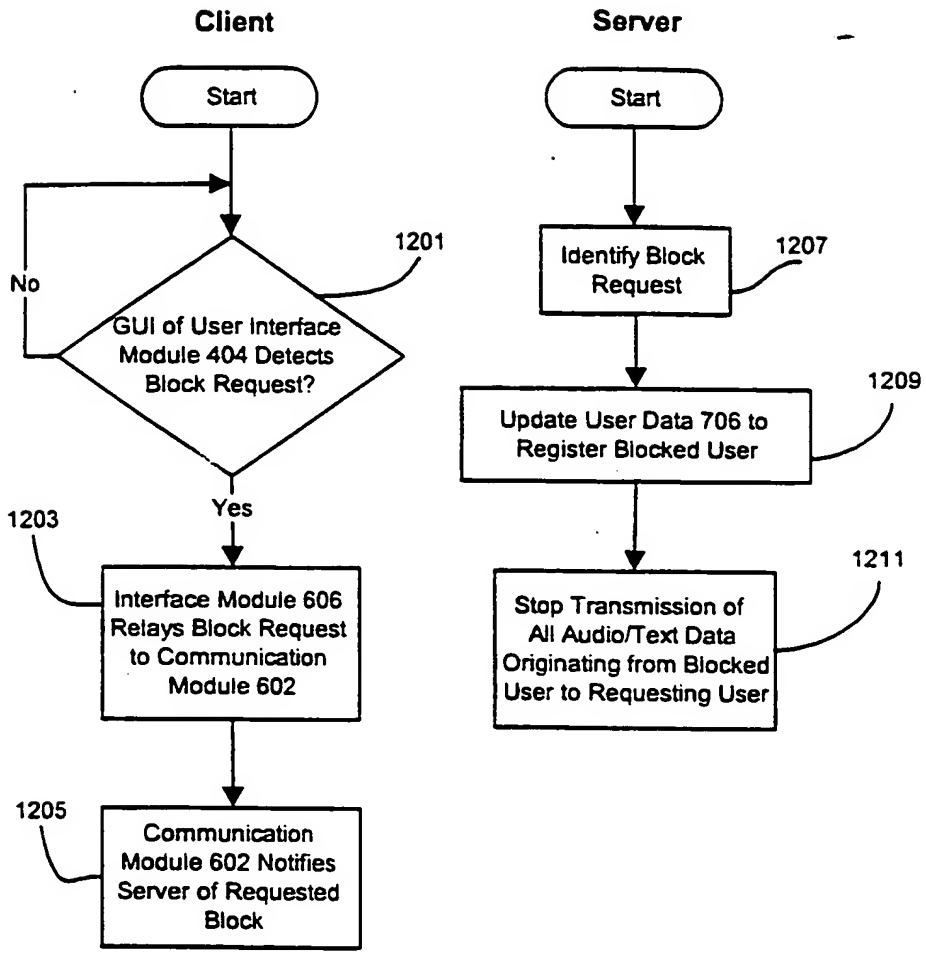

Client
Server
Start
Start
1201
No
GUI of User Interface Module 404 Detects Block Request?
Identify Block Request
1207
1209
Update User Data 706 to Register Blocked User
Yes
1203
Interface Module 606 Relays Block Request to Communication Module 602
1211
Stop Transmission of All Audio/Text Data Originating from Blocked User to Requesting User
1205
Communication Module 602 Notifies Server of Requested Block

Figure 12

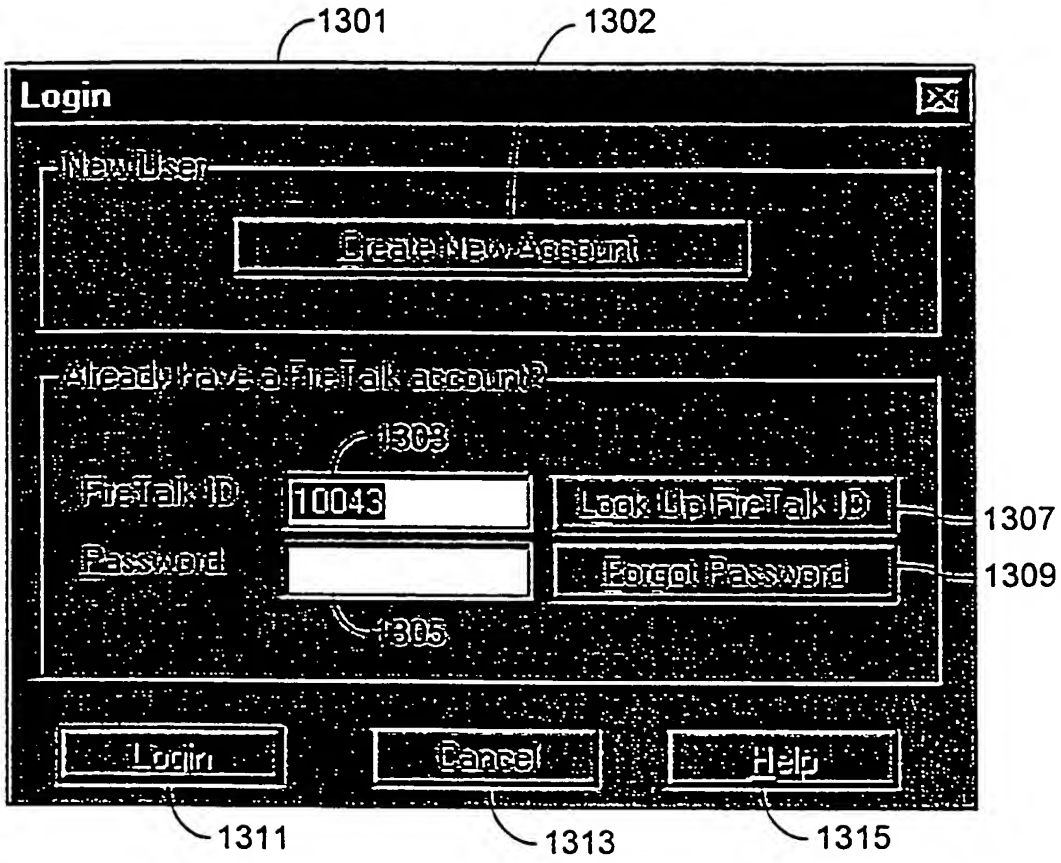

1301
1302
Login
New User
Create New Account
Already have a FireTalk account?
1303
FireTalk ID
10043
Look Up FireTalk ID
1307
Password
Forgot Password
1309
1305
Login
Cancel
Help
1311
1313
1315

Figure 13

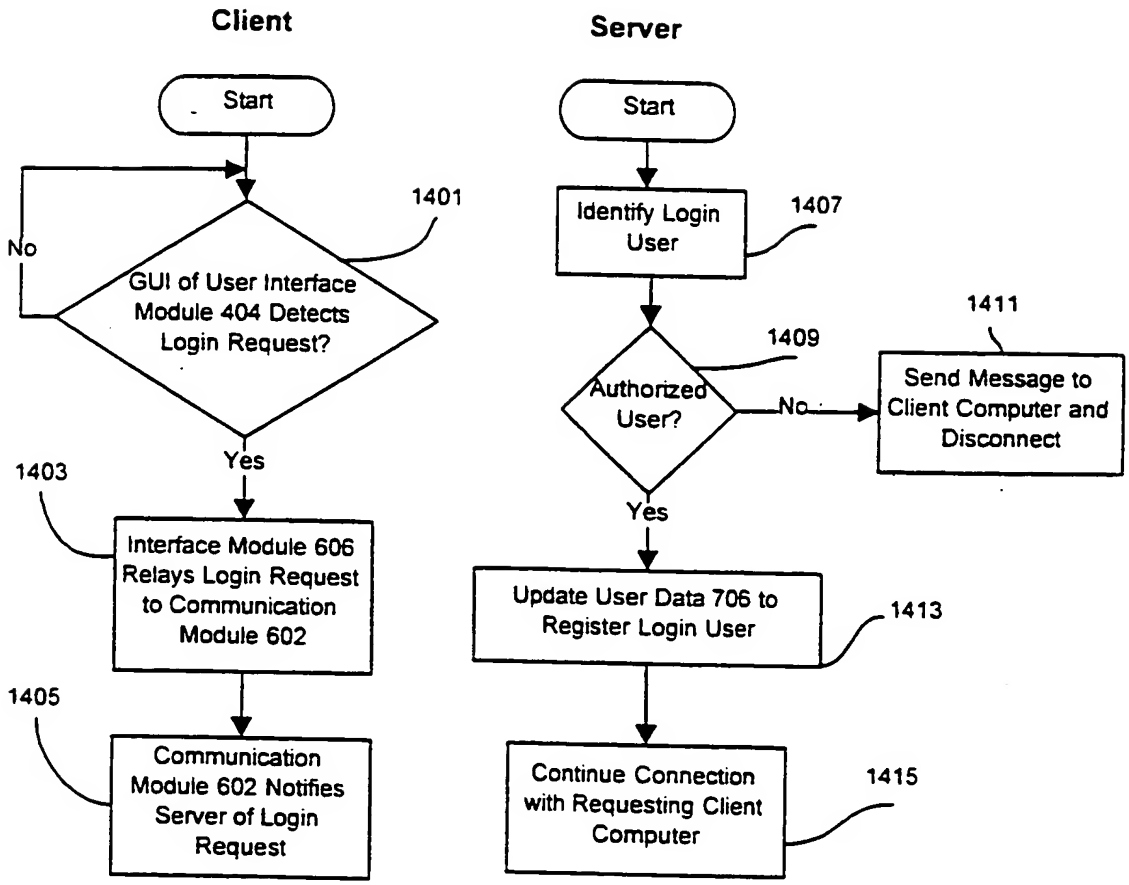

Client
Server
Start
Start
1401
GUI of User Interface Module 404 Detects Login Request?
No
Yes
1403
Interface Module 606 Relays Login Request to Communication Module 602
1405
Communication Module 602 Notifies Server of Login Request
Identify Login User
1407
1409
Authorized User?
No
1411
Send Message to Client Computer and Disconnect
Yes
Update User Data 706 to Register Login User
1413
Continue Connection with Requesting Client Computer
1415

Figure 14

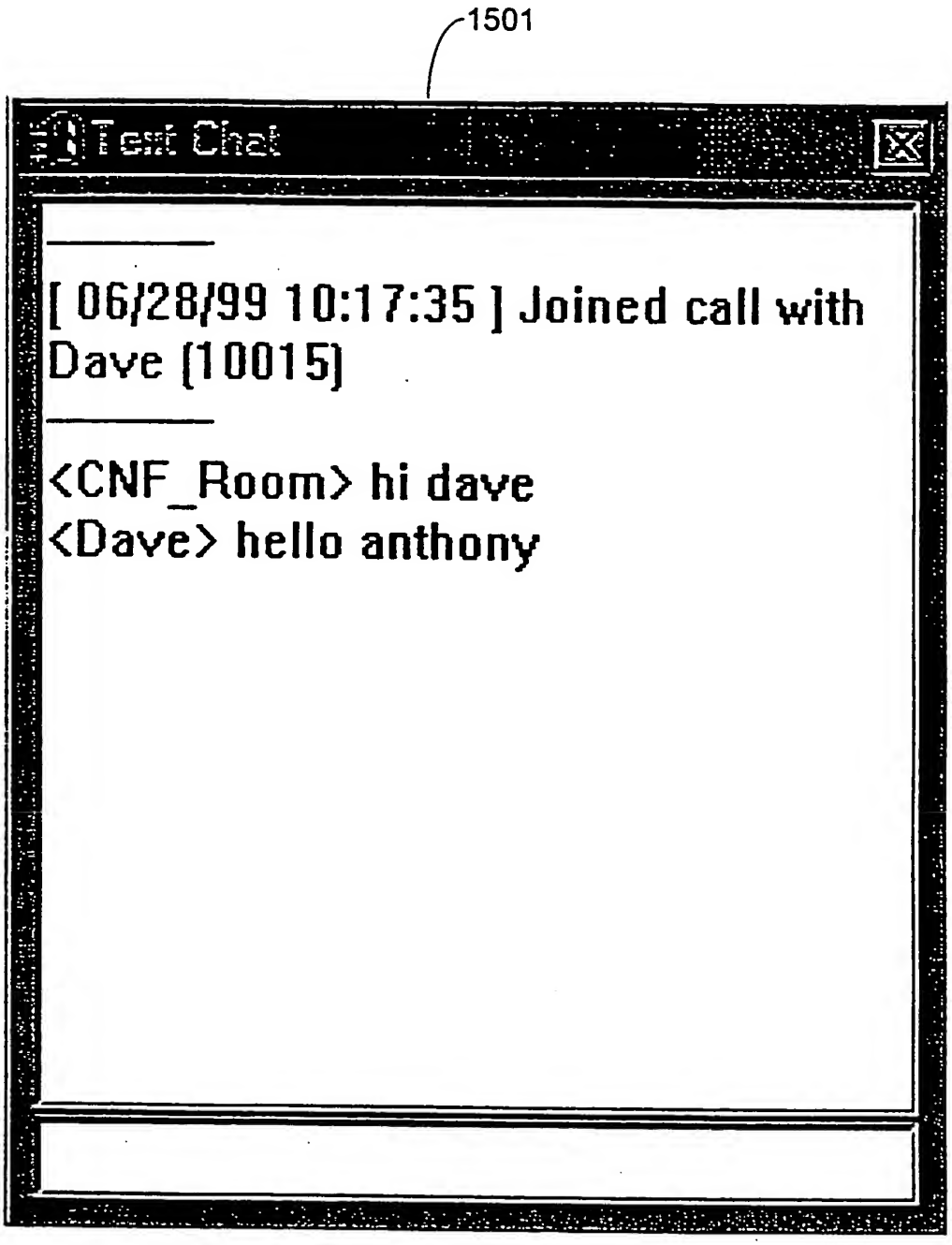

1501
Test Chat
[06/28/99 10:17:35] Joined call with Dave [10015]
<CNF_Room> hi dave
<Dave> hello anthony

Figure 15

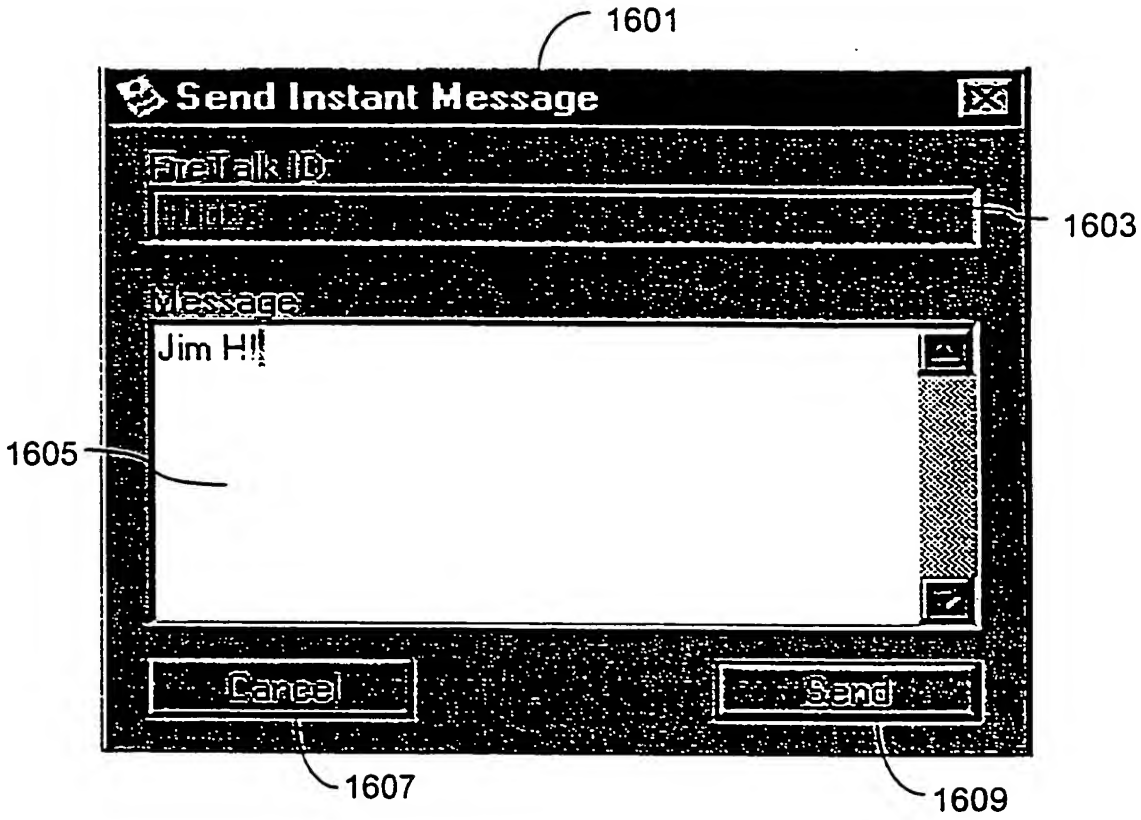

1601
Send Instant Message
FireTalk ID
1603
Message
Jim Hi!
1605
Cancel
Send
1607
1609

Figure 16

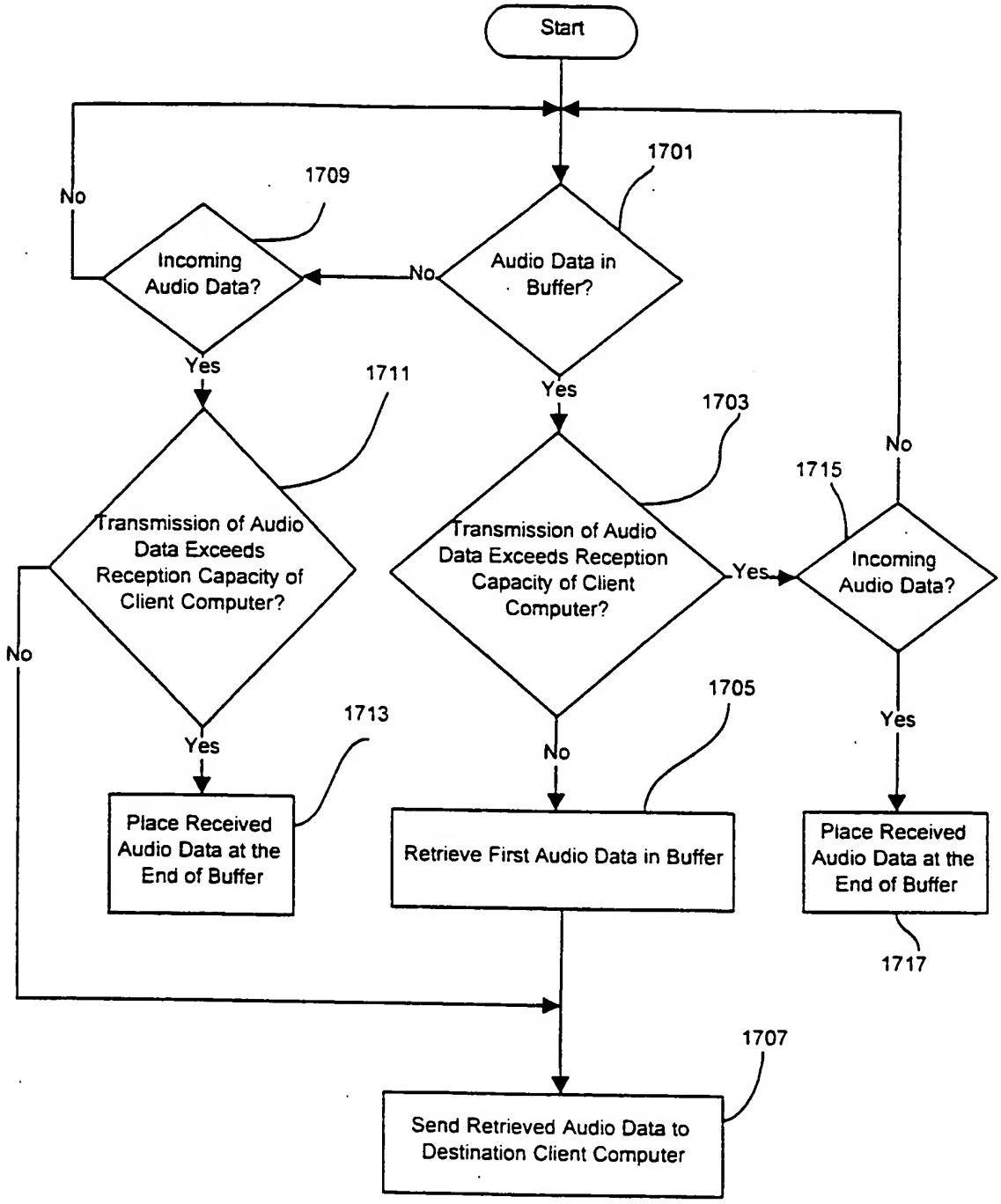

Start
1701
Audio Data in Buffer?
No
1709
Incoming Audio Data?
No
Yes
1711
Transmission of Audio Data Exceeds Reception Capacity of Client Computer?
No
Yes
1713
Place Received Audio Data at the End of Buffer
Yes
1703
Transmission of Audio Data Exceeds Reception Capacity of Client Computer?
Yes
No
1705
Retrieve First Audio Data in Buffer
1715
Incoming Audio Data?
No
Yes
Place Received Audio Data at the End of Buffer
1717
1707
Send Retrieved Audio Data to Destination Client Computer

Figure 17

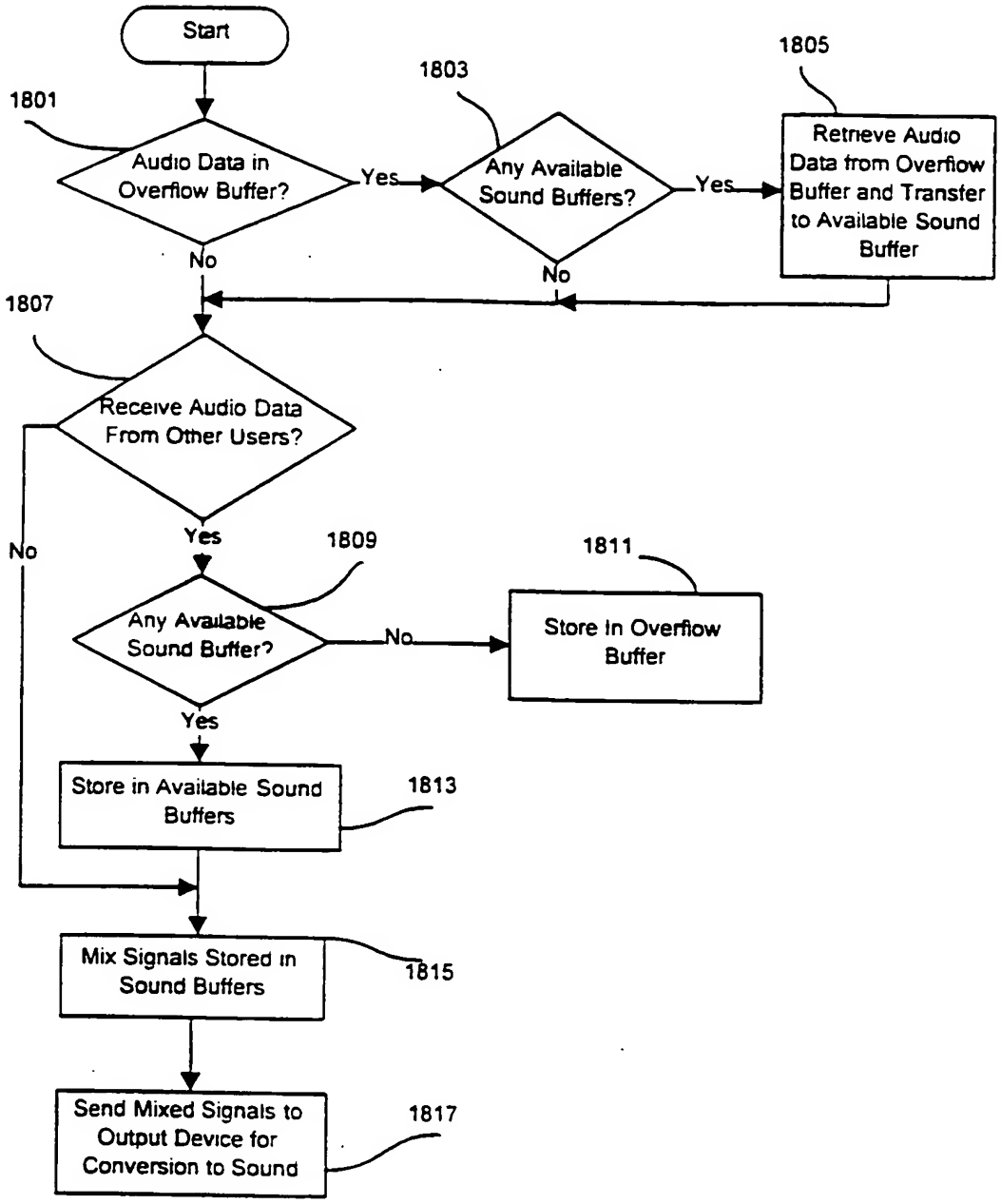

Start
1801
Audio Data in Overflow Buffer?
Yes
1803
Any Available Sound Buffers?
Yes
1805
Retrieve Audio Data from Overflow Buffer and Transfer to Available Sound Buffer
No
No
1807
Receive Audio Data From Other Users?
Yes
No
1809
Any Available Sound Buffer?
No
1811
Store In Overflow Buffer
Yes
Store in Available Sound Buffers
1813
Mix Signals Stored in Sound Buffers
1815
Send Mixed Signals to Output Device for Conversion to Sound
1817

Figure 18

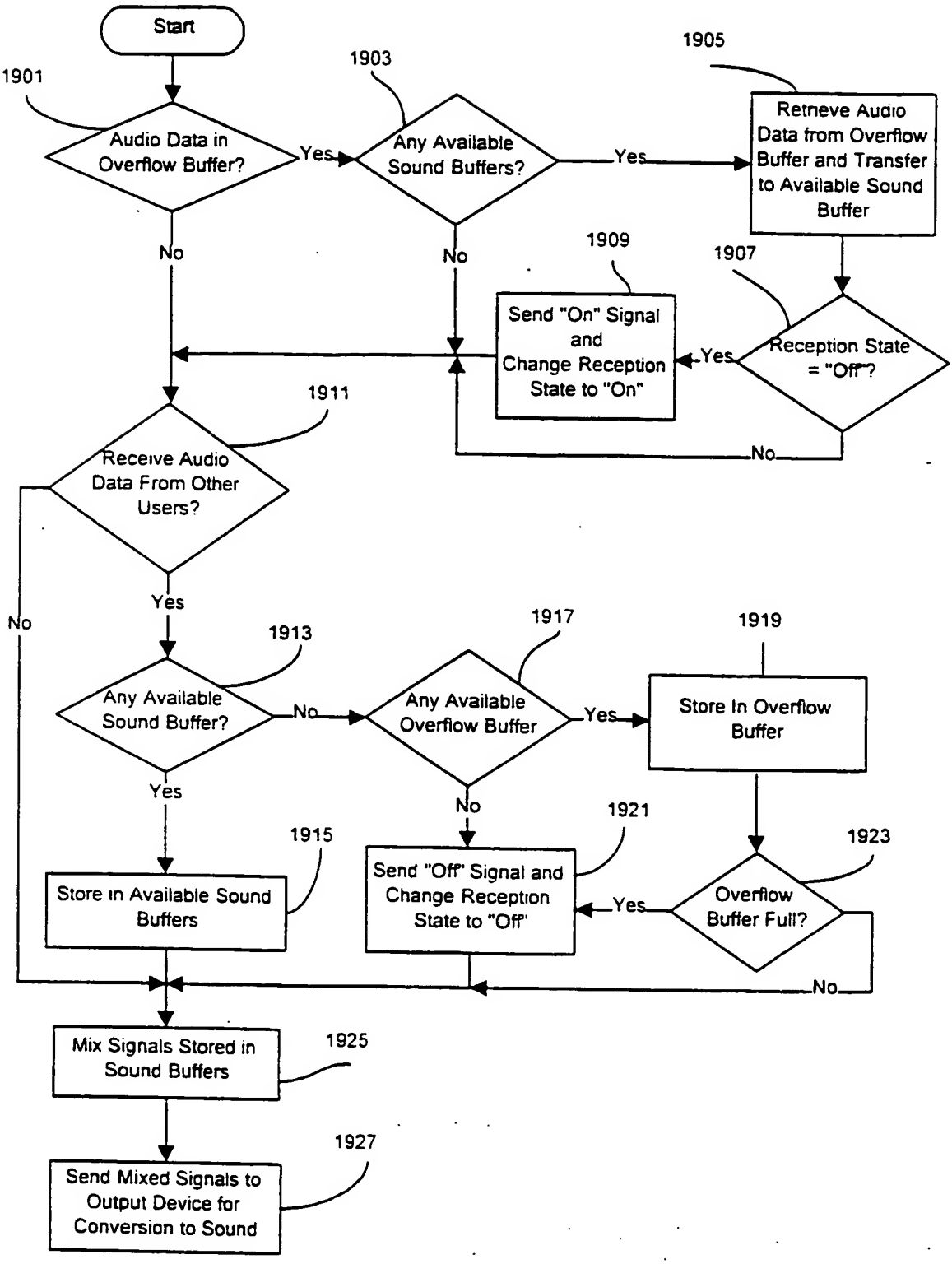

Start
1901
Audio Data in Overflow Buffer?
Yes
1903
Any Available Sound Buffers?
Yes
1905
Retneve Audio Data from Overflow Buffer and Transfer to Available Sound Buffer
No
No
1909
Send "On" Signal and Change Reception State to "On"
1907
Yes
Reception State = "Off"?
No
1911
Receive Audio Data From Other Users?
Yes
No
1913
Any Available Sound Buffer?
No
1917
Any Available Overflow Buffer
Yes
1919
Store In Overflow Buffer
Yes
No
1921
1923
1915
Store in Available Sound Buffers
Send "Off" Signal and Change Reception State to "Off"
Yes
Overflow Buffer Full?
No
1925
Mix Signals Stored in Sound Buffers
1927
Send Mixed Signals to Output Device for Conversion to Sound

Figure 19

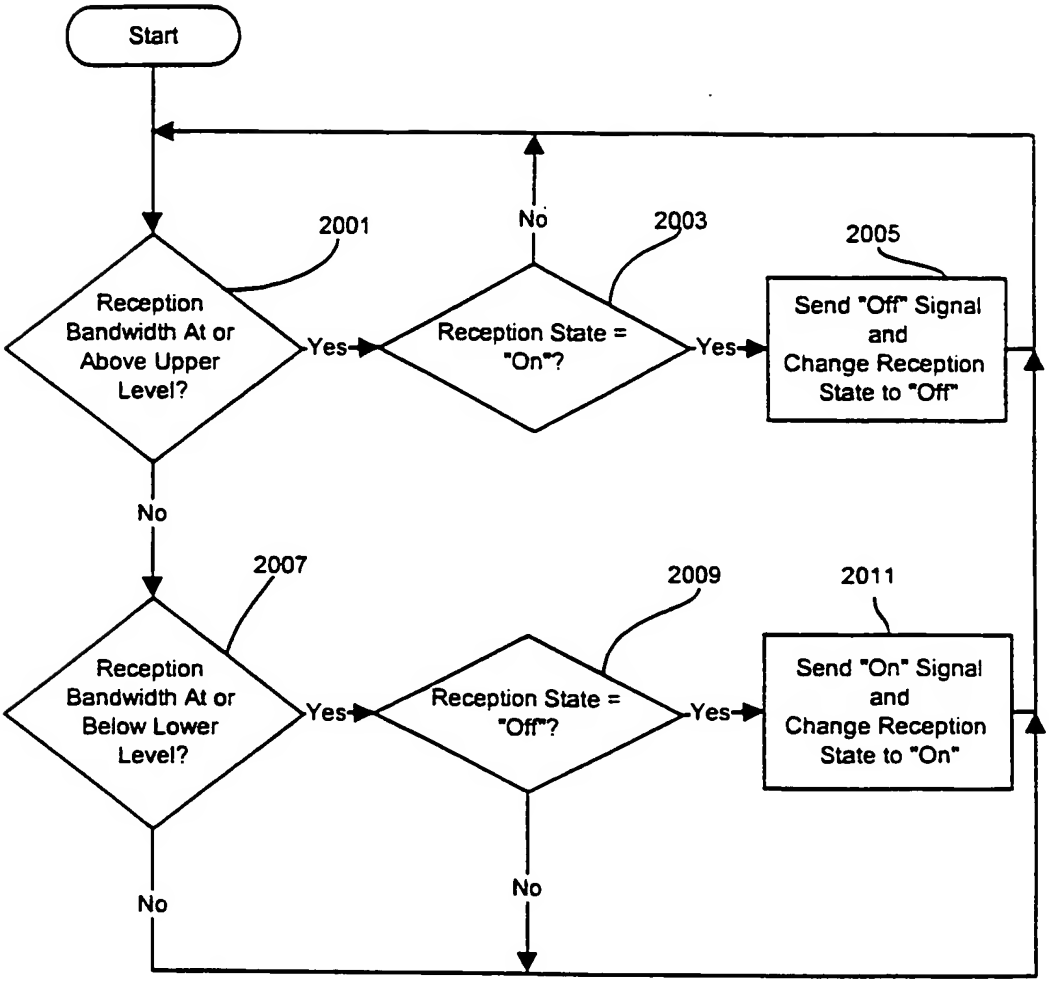

Start
2001
Reception Bandwidth At or Above Upper Level?
Yes
2003
Reception State = "On"?
Yes
2005
Send "Off" Signal and Change Reception State to "Off"
No
No
2007
Reception Bandwidth At or Below Lower Level?
Yes
2009
Reception State = "Off"?
Yes
2011
Send "On" Signal and Change Reception State to "On"
No
No

Figure 20

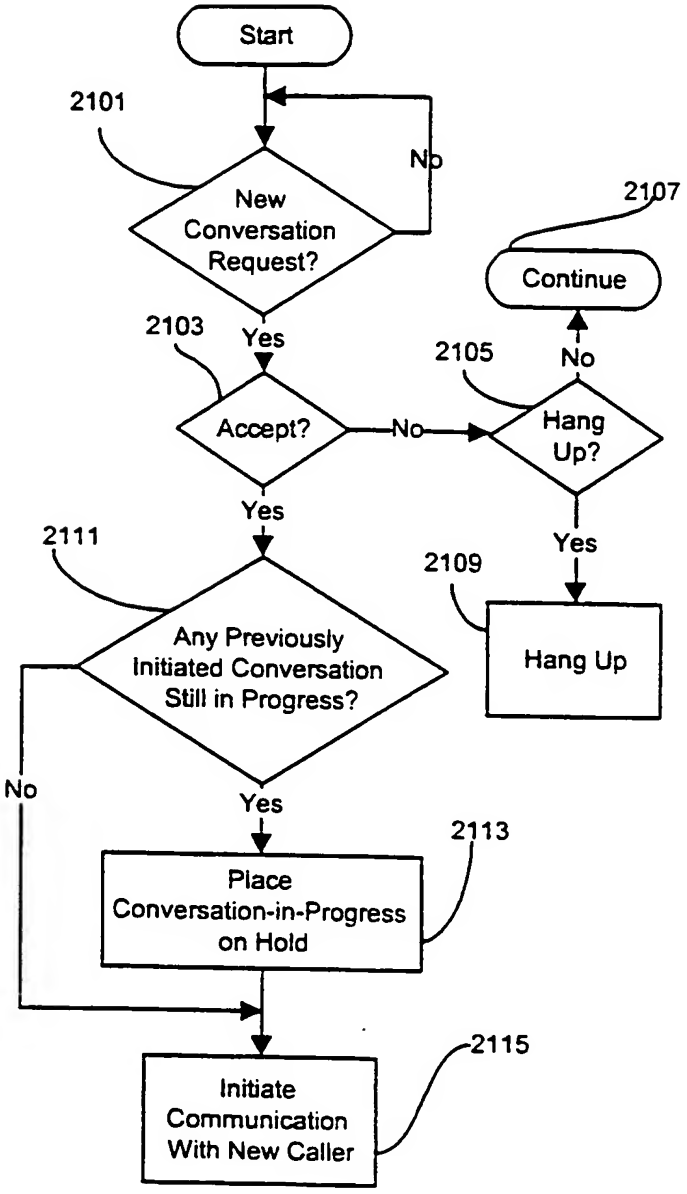

Start
2101
New Conversation Request?
No
Yes
2103
Accept?
No
2105
Hang Up?
No
2107
Continue
Yes
2109
Hang Up
Yes
2111
Any Previously Initiated Conversation Still in Progress?
No
Yes
2113
Place Conversation-in-Progress on Hold
2115
Initiate Communication With New Caller

Figure 21

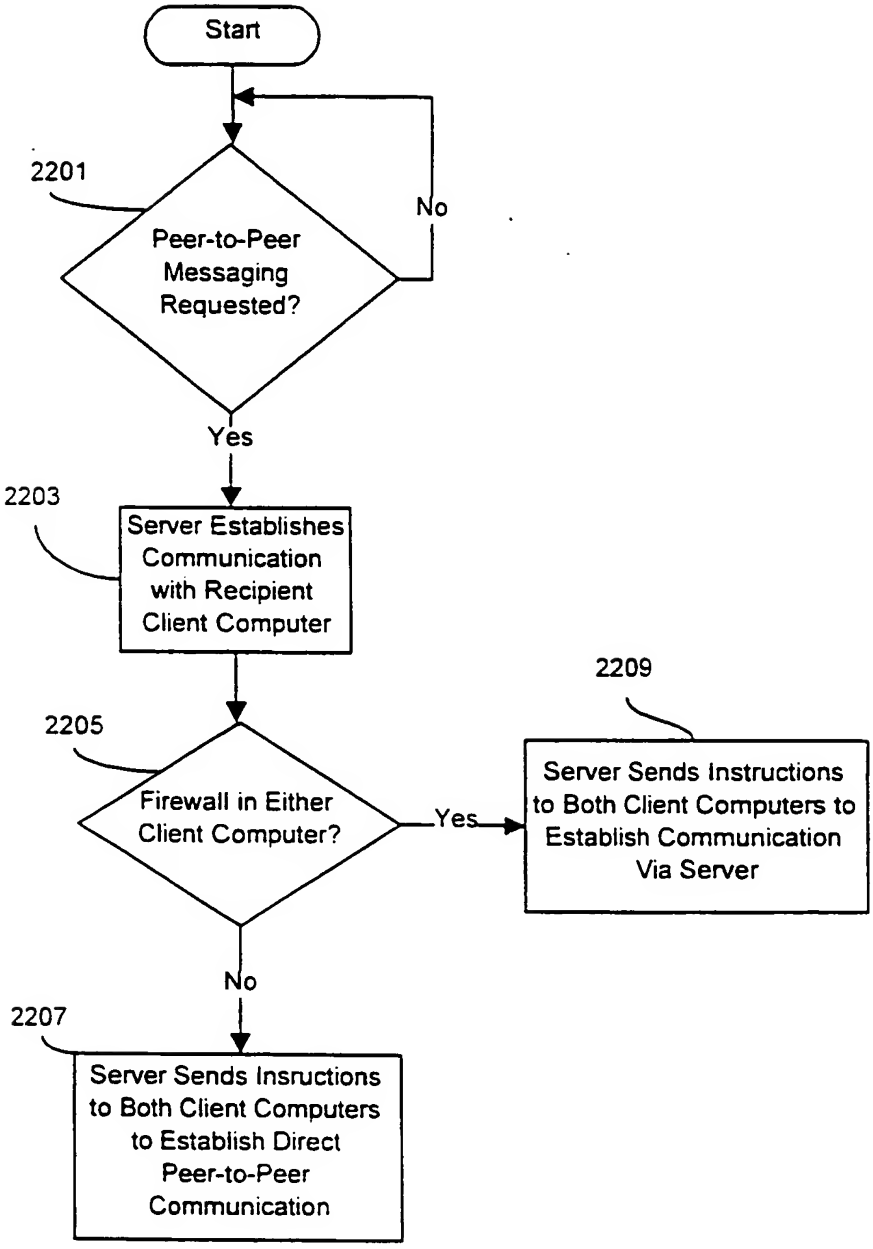

Start
2201
Peer-to-Peer Messaging Requested?
No
Yes
2203
Server Establishes Communication with Recipient Client Computer
2205
Firewall in Either Client Computer?
Yes
2209
Server Sends Instructions to Both Client Computers to Establish Communication Via Server
No
2207
Server Sends Insructions to Both Client Computers to Establish Direct Peer-to-Peer Communication

Figure 22